TechNotes

MARCH 1987

AN ASHTON-TATE
SOFTWARE SUPPORT
PUBLICATION

$6.00

MAR 25 1987

- **A First Look at RapidFile**
- **Change Summary for Framework II Version 1.1**
- **Change Summary for dBASE III PLUS Version 1.1—Part III**
- **Field Ditto During Data Entry**

ASHTON-TATE

SUBSCRIBE TODAY

TechNotes

Don't miss a single issue of Ashton-Tate's hottest source of technical know-how! Each monthly issue of TechNotes provides you with the most current, accurate technical information available from the Software Support Center on all dBASE, Framework, and MultiMate products. Call our toll-free numbers to subscribe: **(800) 331-4164** (in California) or **(800) 828-2514** (residents of other states). In Canada, call **(619) 487-3212.**

☐ Bill me:
Amount Enclosed
☐ $50 U.S.
☐ $65 Canada & Mexico
☐ $75 Other countries

☐ Form of payment Enclosed
☐ Check (Make payable to Ashton-Tate in U.S. dollars drawn on a U.S. bank)
☐ Visa ☐ MasterCard

Expiration Date:

Signature:

☐ New Subscription ☐ Renewal

Name

Address

City

State/Province

Zip/Postal Code

Country

Phone Number

Which product do you use? ☐ dBASE II (D2) ☐ dBASE III (D3) ☐ dBASE III PLUS (DP) ☐ Framework (FW) ☐ MultiMate (MM)
Please allow 30 to 60 days for delivery of your first issue. 1000

E N T S

–Part III 9

13

17

25

Importing and Exporting SYLK Files with Framework II 38
by Quinn Wildman and Karen Robinson

MultiMate

Creating Endnotes 48
by Meryl Horwitz

RapidFile

A First Look at RapidFile 3
An Interview with Mark McDonough and Steve Allin

Departments

Read Me First 1
by Brenda Johnson-Grau

Usage Tips 51

On-Line Services 60

International 64

Letters 69

Index 72

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 66 TORRANCE, CA

POSTAGE WILL BE PAID BY ADDRESSEE

TechNotes Subscription Service
P.O. Box 3729
Escondido, CA 92025-0929

TechNotes

TABLE OF CONTENTS

dBASE III PLUS	Change Summary for dBASE III PLUS Version 1.1—Part III *by Gail Kulikowsky*	9
	Field Ditto During Data Entry *by Daryl Seybold*	13
	Converting Numbers to Words *by Kenneth N. Getz and Karen Robinson*	17
Framework II	Change Summary for Framework II Version 1.1 *by Gail Kulikowsky*	25
	Importing and Exporting SYLK Files with Framework II *by Quinn Wildman and Karen Robinson*	38
MultiMate	Creating Endnotes *by Meryl Horwitz*	48
RapidFile	A First Look at RapidFile *An Interview with Mark McDonough and Steve Allin*	3
Departments	Read Me First *by Brenda Johnson-Grau*	1
	Usage Tips	51
	On-Line Services	60
	International	64
	Letters	69
	Index	72

READ ME FIRST

BY BRENDA JOHNSON-GRAU

Let's see what the March wind has blown into this issue. Karen Robinson and Gail Kulikowsky interviewed Mark McDonough and Steve Allin of the RapidFile development team. "A First Look at RapidFile" is their overview of Ashton-Tate's new file management software product. Gail Kulikowsky also wrote Part III of the "Change Summary for dBASE III PLUS Version 1.1" and a "Change Summary for Framework II Version 1.1."

"Importing and Exporting SYLK Files with Framework II," by Quinn Wildman and Karen Robinson, continues our effort to provide programming solutions that allow you to use files produced by other software products. These programs can be used to import and export SYLK files produced by Microsoft's Multiplan—either from the Import and Export Add-In options on the Disk menu or as macros.

Daryl Seybold has written a handy program for users that need a field, rather than a record, ditto in dBASE III PLUS. With the program supplied in "Field Ditto During Data Entry," you can carry forward the data from the previous record to the current field by simply pressing a function key.

"Converting Numbers to Words" reveals our untiring assembly language programmer Kenneth N. Getz using the LOAD and CALL commands to convert numbers into words expeditiously—with color commentary by Karen Robinson.

In "Creating Endnotes," Meryl Horwitz shows you how to use the UTIL file conversion utility to collect the footnotes in a MultiMate Advantage document together as endnotes.

Alistair Connor and Didier Rochereau of La Commande Electronique, the Ashton-Tate office in France, provide instructions on using Framework II to emulate a Minitel. The Minitel is a popular dumb terminal used to access the many electronic bulletin boards available in France.

This month's *Usage Tips* include some for *dBASE Tools for C*. Of special interest among the dBASE III PLUS tips are a program for creating something (that is, a database file) from nothing and a program to convert numbers in decimal notation to scientific notation. The auto indent macro is a practical aid for programmers who use Framework II as a program editor and who would like to spend less time indenting code lines.

READ ME FIRST

BY BRENDA JOHNSON-GRAU

Let's see what the March wind has blown into this issue. Karen Robinson and Gail Kulikowsky interviewed Mark McDonough and Steve Allin of the RapidFile development team. "A First Look at RapidFile" is their overview of Ashton-Tate's new file management software product. Gail Kulikowsky also wrote Part III of the "Change Summary for dBASE III PLUS Version 1.1" and a "Change Summary for Framework II Version 1.1."

"Importing and Exporting SYLK Files with Framework II," by Quinn Wildman and Karen Robinson, continues our effort to provide programming solutions that allow you to use files produced by other software products. These programs can be used to import and export SYLK files produced by Microsoft's Multiplan—either from the Import and Export Add-In options on the Disk menu or as macros.

Daryl Seybold has written a handy program for users that need a field, rather than a record, ditto in dBASE III PLUS. With the program supplied in "Field Ditto During Data Entry," you can carry forward the data from the previous record to the current field by simply pressing a function key.

"Converting Numbers to Words" reveals our untiring assembly language programmer Kenneth N. Getz using the LOAD and CALL commands to convert numbers into words expeditiously—with color commentary by Karen Robinson.

In "Creating Endnotes," Meryl Horwitz shows you how to use the UTIL file conversion utility to collect the footnotes in a MultiMate Advantage document together as endnotes.

Alistair Connor and Didier Rochereau of La Commande Electronique, the Ashton-Tate office in France, provide instructions on using Framework II to emulate a Minitel. The Minitel is a popular dumb terminal used to access the many electronic bulletin boards available in France.

This month's *Usage Tips* include some for *dBASE Tools for C*. Of special interest among the dBASE III PLUS tips are a program for creating something (that is, a database file) from nothing and a program to convert numbers in decimal notation to scientific notation. The auto indent macro is a practical aid for programmers who use Framework II as a program editor and who would like to spend less time indenting code lines.

TechNotes

Publisher
Ashton-Tate Support Services
Jay Hanson, Director
Laurie Miller, Manager

Editor-in-Chief
Brenda Johnson-Grau

Editor
Christopher White

Technical Writer
Karen Robinson

Production Coordinator
Olivier Biggerstaff

Technical Staff
Larry Abare, Richard Augelli, Stanley Ballenger, Rodney Barbati, Richard Bunter, Mike Carlisle, Fred Crownover, Michael Fiore, Rick Fillman, Charles Gehman, Ken Getz, Richard Goepel, Peter Gold, Mina Gussin, Doug Haman, Scott Haynes, Mike Hedblom, Clint Helton, Errol Jackson, Matt Johnson, Rosaline Keenan, Gail Kulikowsky, Steve Kurasch, Jerry Levy, Charles Litzell, Chris Lo Vasco, Jeffrey McCrimon, Jerry Mendes, Vincent Mendillo, Elizabeth Miranda, Jeffrey Moore, Keith Mund, Joseph Perret, Jaspal Rattan, John Regen, Deborah Ridolfo, Daryl Seybold, Cathy Shikatani, Steve Silverwood, Joseph Stolz, Naoma Stow, Philip Talsky, Mitra Toossi, Hiram Vega, Ralph Wakefield, Lisa Wanek, Mark Ware, Quinn Wildman, Roger Wegehoft

Contributing Editors
Paul Bukauskas (International), Liz Burke (MultiMate), Steve Silverwood (On-Line Services)

Administrative Assistant
Kathy Smith

Illustrator
Stephen Goodman

Production Staff
Gary Ashcavai, D. A. Gray Bare, Alecia C. Bernard, Tim Dunne, Jeff Elliott, Darrell Evers, R. James Landers, Renée Maynard, Patty Pesek, Ron Powell, Ann O. Taono, Ruth J. Velazquez

Product Development
Forefront Development Center: Marilyn McMahon, Greg Stikeleather
Glendale Development Center: Debby Moody

DISCLAIMER NOTICE

The information contained in this publication is designed to help you develop and improve your programming skills and to assist you in using Ashton-Tate products. Ashton-Tate assumes no responsibility whatsoever for the uses made of any of the software code in this issue, whether modified or unmodified.

If you discover what you believe is an anomaly (as defined under **Anomalies**) not presented in this issue or others, please send a detailed description of the possible anomaly to:

Ashton-Tate Software Support Center
20101 Hamilton Ave.
Torrance, CA 90502-1319

If the reported anomaly can be reproduced by our technicians, it may be added to a future issue of ***TechNotes***.

The policies that are described in ***TechNotes*** apply only to the United States (and possessions), Canada, and Mexico. Residents of other foreign countries should contact a local Ashton-Tate representative in their country for policy information and technical support.

Ashton-Tate and the authors will not be liable for special, incidental, consequential, indirect or other similar damages, even if we have been advised of the possibility of such damages. This means we are not responsible for damages or costs incurred as a result of loss of time, loss of data, loss of profits or revenues, or loss of use of the software. In addition, we are not responsible for damages or costs incurred in connection with obtaining substitute software, claims by others, inconvenience or similar costs.

No warranties, either expressed or implied, are made regarding the contents of this publication, its merchantability or fitness for any particular purpose, all of which are hereby expressly disclaimed. Note, however, that some states do not allow the exclusion or limitation of warranties or of direct, incidental or consequential damages, so these warranty limitations may not apply to you.

In no event will our liability for any damages to you or any other person ever exceed the price paid for your subscription to ***TechNotes***, regardless of any form of the claim.

Additional statements by agents, employees, distributors and dealers of Ashton-Tate do not constitute warranties by Ashton-Tate and do not bind Ashton-Tate.

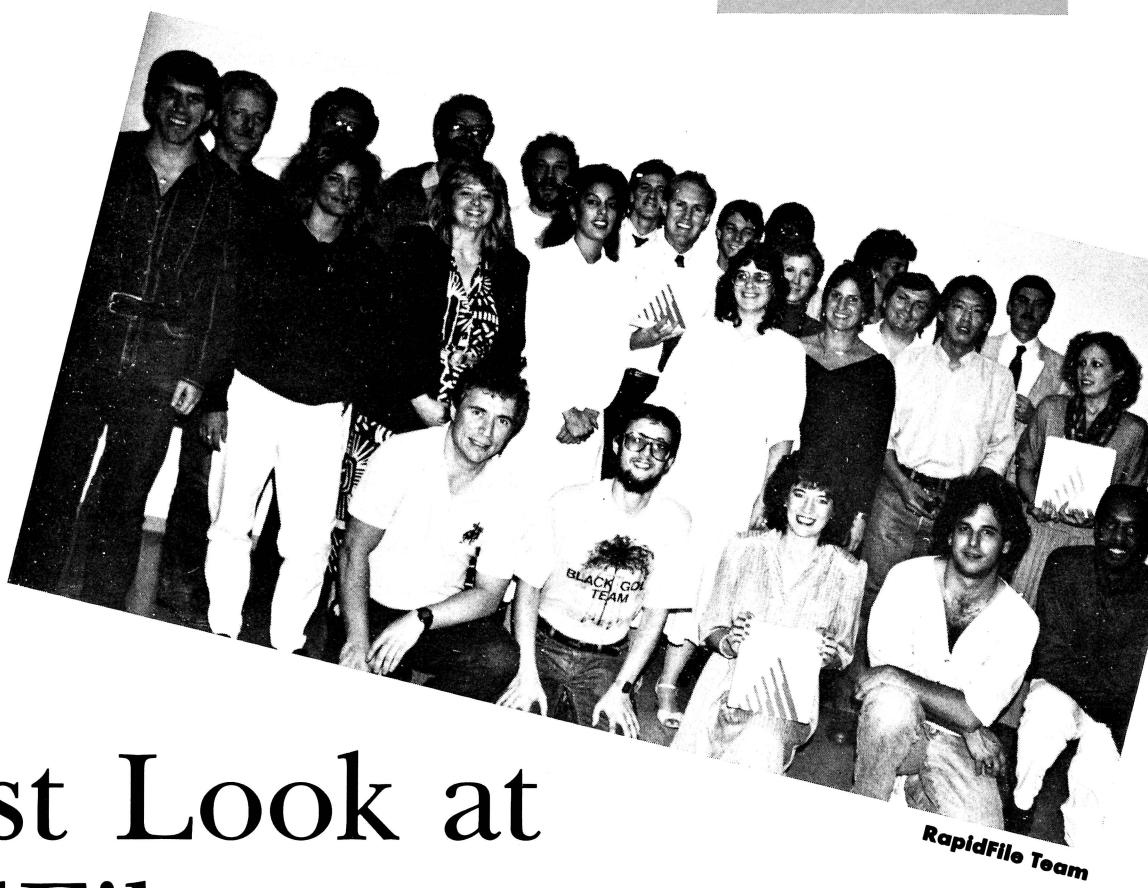
TechNotes (ISSN 0883-6493) is an Ashton-Tate Software Support Center publication containing technical reference information on Ashton-Tate and MultiMate products.

Entire contents Copyright © Ashton-Tate 1987. All rights reserved. Commercial reproduction of the material appearing in ***TechNotes*** is forbidden without prior written permission. Direct all requests for reproduction permission to Ashton-Tate Support Services, 20101 Hamilton Ave., Torrance, CA 90502-1319 ATTN: Brenda Johnson-Grau.

TechNotes is published monthly at a subscription price of $50.00 for one year. Additional rates: Canada and Mexico, $65.00; other countries, $75.00. Monthly issues are available in back issues for $10.00 each (1985 issues) and $7.00 each (1986 and 1987 issues). ***TechNotes*** is published by Ashton-Tate, 20101 Hamilton Ave., Torrance, CA 90502-1319. POSTMASTER: Send subscription inquiries and address changes to ***TechNotes*** Subscription Service, P.O. Box 3729, Escondido, CA 92025-0929.

dBASE II, dBASE III PLUS, and Framework II are registered trademarks of Ashton-Tate. dBASE, RunTime, FRED, and ASHTON-TATE are trademarks of Ashton-Tate. Some of the trademarks of other companies mentioned in this issue appear for identification purposes only.

The Ashton-Tate Software Support Center's hours are 6:00 a.m. to 4:30 p.m., PST, Monday through Friday, except on Thursday when we close at 4:00 p.m. To reach our technicians for technical support, call (213) 329-0086. Be sure to have your software serial number ready for our operators.



RapidFile Team

A First Look at RapidFile

An Interview with Mark McDonough and Steve Allin

RapidFile™ is Ashton-Tate's newly released data management product for the IBM PC and 100% compatibles. It's a fast, easy-to-use file manager that combines file management, reporting, and form letter capabilities on one disk. Karen Robinson of ***TechNotes*** and Gail Kulikowsky recently talked with Mark McDonough, the developer, and Steve Allin, the RapidFile Technical Product Manager. The interview was interesting and informative. We got an inside view on why RapidFile was developed, the developmental paths it took, and what it does.

TN: How did RapidFile get started?

MM: When I was at MIT's Business School, I was inspired by an IBM product called QBE (Query by Example). Its visual interface offered more power to database users without sacrificing ease of learning. I sketched out the original design, pulled together the money, and created a team of highly talented programmers to implement it. Since that time, the product has gone through many incarnations and has benefited from the suggestions of many creative software designers as well as from the user research conducted in Ashton-Tate's test labs. The most important and significant changes to the design came from Steve Allin's work; he brought consistency to the interface.

TN: How about you, Steve? What was your role?

SA: My job was basically to manage the technical aspects of the software development process. Inevitably I became very involved with the design. As Mark said, many people had a hand in creating RapidFile, which provided a lot of good ideas. For a while though, the design started getting pulled in several different directions. I just took all the pieces and put them together so that they made sense.

TN: Tell us about RapidFile. What is it?

MM: RapidFile is a low-end file manager intended for the novice user who has single file applications. The term "low-end" sometimes suggests that the product is a "toy" or not serious—not true with RapidFile. RapidFile is a very powerful file management product for single file applications that is easy to use. We use the term "low-end file manager" to make a distinction between a personal productivity tool and a full-blown relational database system such as dBASE III PLUS. We expect people to use RapidFile to keep track of data such as client lists rather than to build full applications such as accounts receivable systems.

SA: Users are getting more sophisticated today. RapidFile is meant to grow with its users. It does the basic tasks very well and very easily, but it doesn't stop there. You can use calculated fields and report fields to build advanced applications that can't be created in other flat file programs.

TN: What is a flat file?

SA: A flat file is a file which isn't related to other files. It's difficult to explain without getting into a description of relational databases, but in the simplest terms a flat file stands alone. A simple mailing list is an example of a flat file.

TN: Give us an overview of RapidFile.

MM: RapidFile is a complete file manager. It includes functions for managing data, printing reports, and producing mailing labels and form letters. It has six different types of layouts for browsing and editing data. Using the layout and snapshot functions, you can save hundreds of templates under unique names.

It has report writing capabilities that range from columnar reports with simple calculations to custom reporting. Custom reports provide the power and flexibility for creating free-form reports that can print to pre-printed forms or checks.

RapidFile's memo writer offers an easy approach to doing business form letters. You can address a letter to many people from the data file, or just one. The memo writer allows you to quickly merge data from the data file into a letter, without retyping any information. You can also see the format of the letter before printing it.

TN: Why did you develop RapidFile?

MM: We developed RapidFile because we wanted to address the needs of users who see a gap between the very sophisticated database managers (such as dBASE III PLUS) and the very simple flat file management products now in the market place. We wanted to develop a product that would be easy to use and still go beyond the capabilities of those low-end file managers.

The problem with most low-end file managers is that you eventually "hit a wall." For a time you could be happily going along using the product and then want to do one more data manipulation task only to discover that there is no feature to do it. We really tried hard to design RapidFile so that data management tasks could move from simple to more complex all in one product. It has an open-endedness that allows you to perform tasks that are impossible to do with other file managers.

RapidFile was designed so that tasks done 80% of the time are very easy to use, and tasks done 20% of the time, which are usually the more advanced tasks, are hidden behind the scenes so as not to confuse the new user. This layering of capabilities makes it easy to start using the product, while giving more power to the users when they need it.

TN: Tell us about the 80%.

SA: Our primary emphasis was on the applications that people do most commonly: labels, form letters, reports, and of course data management. We set out to make a product that could handle these tasks very easily and with more flexibility than other file managers. There are many users with very simple file management needs. So, our first goal was to give these users simple solutions.

TN: How about the 20%?

MM: A good example of layered functionality are the two report writers. We have the quick report which falls into the 80% and gives you the simple subtotals and totals, averages or whatever, and then the custom report which falls into the 20% and offers complete flexibility in designing layouts and statistical calculations. Designing a custom report is certainly more involved (no pain, no gain), but it's not required. You don't have to master all of RapidFile's reporting possibilities in order to print a simple report.

TN: So, you're not competing with dBASE.

MM: No, we're not. It's important to understand the different functions of RapidFile and dBASE III PLUS. RapidFile was specifically designed as a personal productivity tool. It is a fast, single file manager with excellent reporting capabilities. It has macros, and will allow users to access data by means of a network, although users cannot update a file simultaneously as with dBASE III PLUS. We see RapidFile competing quite successfully with PFS:file® and Q&A.

dBASE III PLUS, on the other hand, offers relational capabilities, a programming language, and full multi-user functionality. With RapidFile, we expect to serve users that Ashton-Tate has not always served before. And because Ashton-Tate now offers a family of products that can share data, users can move between products quite easily.

TN: How does RapidFile compete with PFS®?

SA: It's faster and a lot more flexible. One of the keys to being successful in the software market is ease-of-use. PFS achieved this by limiting its capabilities. We didn't want to do that. As I said before, users are becoming more sophisticated. We didn't want to create a product that users would quickly outgrow. Our goal with RapidFile was to give users capability *and* ease-of-use. Q&A and Reflex™ are both good products, but they have weaknesses. Q&A has many of the same limitations as PFS and requires 512K of RAM, whereas Reflex is limited to RAM for storing data, lacks a word processor, and requires a graphics card. I think our edge is that RapidFile does what most users want without the trade-offs.

MM: Another problem with PFS and Q&A is their interface. It's ancient; it goes back to mainframes. Their report interface still isn't visual. You are given a list of fields and must type in instructional commands for which fields to include, how to sort them, and which to total or subtotal. Only when the report is printed do you see its final format. RapidFile, on the other hand, has a visual report writer. Like a spreadsheet, you can see the report as you design it on the screen.

TN: What features do we offer that PFS Professional package doesn't offer?

SA: We don't have that much time. No really, PFS is a good product, but it isn't very flexible. I guess I'd have to say our strongest features are the fully customizable layouts, calculated and report fields, and the ability to save layouts and snapshots. These features put the user in control; that is, we give you the tools to tailor the product to suit your application rather than requiring you to tailor your application to the product. With layouts for example, PFS provides only one way to browse data. RapidFile provides six different types of layouts from which you can view or print data. Users have different needs, some want to view data in a table, others want the screen display to emulate the pre-printed forms they are entering data from.

TN: Can you explain what layouts and snapshots are?

SA: A layout defines which fields display and in what arrangement. A snapshot ties a layout to a sort order and a selection criteria, giving you the ability to save and recall different views of the data. Building a snapshot is a very natural process, you create it piece by piece: the layout, the sort, and the record selection. When you like what you see, you take (that is, save) the snapshot and every time you use that snapshot, everything is done for you. It's like a template.

TN: Is that how you expect users to build applications?

MM: Yes, snapshots, report fields, and macros are our means of "programming by example." Using these function you can build applications without using a programming language. RapidFile applications will be similar to the income statement and balance sheet spreadsheet templates that allow you to plug in your own numbers. Users can combine snapshots with macros to create template applications.

TN: Macros?

MM: Yes, RapidFile has its own built in macro processor that lets you create your own set of macros.

TN: And RapidFile runs in a 256K environment?

SA: That's right, it runs on an IBM PC or 100% compatible with two floppy drives and 256K of RAM.

RapidFile was originally a memory-based product, which means that the entire file was kept in RAM. To expand the storage capacity, we implemented virtual memory, which uses the disk to extend or simulate memory.

TN: So it uses whatever RAM is available and then goes to the disk?

SA: Yes. Virtual memory allows us to go beyond the 640K RAM barrier and extend the file sizes to several megabytes. As I said, it uses the available disk space to extend the memory. This is especially useful on hard disk systems. We know that many users have mailing lists with over 30,000 names to maintain and we didn't want to limit them.

TN: What about upward compatibility? I mean, right now I just want to build a flat file database. Then later, I want to program an application or define relations between files in dBASE III PLUS. How can I move my RapidFile data into dBASE III PLUS?

MM: It's easy to exchange data between RapidFile and dBASE III PLUS; it literally takes four keystrokes to create a dBASE III PLUS file. In fact, we've gotten reports from users that say they keep their records in dBASE III PLUS, but use RapidFile for fast reporting. It's like having the best of both worlds when you can use dBASE III PLUS to maintain your corporate database and use RapidFile to view the data and create simple reports. RapidFile is like a spreadsheet-database that gives you a visual orientation.

TN: Why is RapidFile written in FORTH?

MM: I'll tell you the story. When I told my professors that I wanted to implement the QBE concept on a PC, I was told it was not possible. Knowing that it couldn't be done, I went out to find a genius that would know how to push the PC to its limits. I found that genius in Tom Dowling.

Tom claimed, and has since proven time and again, that FORTH was as easy to program in as other high level languages and that its programs run almost as fast as assembler programs. FORTH programs also compile into a very small space. We could not have fit RapidFile onto one disk and in 256K of RAM if it were written in C.

TN: We understand that the flexibility of FORTH really came in handy when designing RapidFile.

SA: Yes, FORTH is very tailorable. It allowed us to make substantial changes to the design. It is very close to being a prototyping language where you can make major changes much more easily than in standard programming languages, which would have required substantial rewrites. RapidFile went through several versions before its release and it couldn't have gone through all that if it hadn't been written in FORTH.

MM: Having gone through two publishers before arriving at Ashton-Tate, RapidFile began to resemble a sprawling Victorian mansion built with several competing designs. Steve took RapidFile's "rooms" and rearranged them into a unified whole. We couldn't have done that kind of rearranging without FORTH. We were also able to do all that with only three programmers: Tom Dowling for the major database work, Bent Schmidt-Nielsen for the word processor and system issues, and Dave Fristrom of Eastgate Systems for the interface programming. By using FORTH, we were able to rearrange RapidFile until it was just the way we wanted it.

TN: Well, thank you Steve and Mark for talking with us.

Some of the product names mentioned herein are used for identification purposes only and may be trademarks of their respective companies.

Change Summary for dBASE III PLUS Version 1.1—Part III

BY GAIL KULIKOWSKY

This is the third part of a series listing the anomalies for dBASE III PLUS version 1.0 that were fixed in dBASE III PLUS version 1.1. They are listed here so that dBASE III PLUS version 1.0 users get a clear idea of which anomalies have been corrected and which are still outstanding. This list includes anomalies fixed in both dBASE III PLUS and in the dBASE Administrator.

dBASE Administrator

CONFIG.DB / DEFAULT

In the dBASE Administrator version 1.0, if DEFAULT was SET TO another drive in CONFIG.DB or in the first called .PRG, trying to load the Access disk caused the error message "Unauthorized duplicate" to display and control returned to the DOS prompt.

In version 1.1 of the dBASE Administrator, this no longer happens.

DIR

In the dBASE Administrator version 1.0, DIR on read/only (R/O) files returned the error message "not readable."

In version 1.1, you can use DIR to get a directory listing on read-only files without error.

INDEX / EXCLUSIVE

In the dBASE Administrator version 1.0, USEing a database file with two open index files and SETting EXCLUSIVE OFF may have corrupted the first index file.

Now in version 1.1 of the dBASE Administrator, you can USE a database file with two index files without index corruption of the second one.

INDEX / PC-NET

In the dBASE Administrator version 1.0, creating multiple index files with a large database file (7,000 or more records) may have caused your computer to freeze.

In version 1.1 of the dBASE Administrator, you can create multiple index files with large database files successfully.

MODIFY STRUCTURE

In the dBASE Administrator version 1.0, MODIFY STRUCTURE generated the error message "NET 803: Network path not found" and left zero records in the database file if the current directory of the drive was the system's physical or logical root directory.

In dBASE Administrator version 1.1, you can MODIFY STRUCTURE of a database file in the root directory without error and damage.

PROTECT

In the dBASE Administrator version 1.0, Protect sometimes created encrypted .CRP files with embedded end-of-file characters. The original .DBF file, however, was not affected. At times, it appeared that encrypting a 1,000 record database file created a .CRP file with between one and four embedded end-of-file characters.

In version 1.1 of the dBASE Administrator, you can encrypt database files with Protect in excess of 1,000 records successfully.

REPLACE / PC-NET

In the dBASE Administrator version 1.0, doing a REPLACE to a non-key field in a database file with two open indexes and SET EXCLUSIVE OFF may have caused the computer to freeze. For this to have happened, the index files must have had key expressions of differing lengths.

In version 1.1 of the dBASE Administrator, you REPLACE to non-key fields without causing the computer to freeze.

SORT

In the dBASE Administrator version 1.0, performing a SORT on a large database (7,000 or more records) may have returned the error message "Records do not balance (program error)" and created a target database file containing garbage characters.

In version 1.1 of the dBASE Administrator, you can now SORT large database files, creating a target database file as expected.

dBASE III PLUS

CREATE / MODIFY SCREEN

In dBASE III PLUS version 1.0, if you filled the blackboard and then pressed **Ctrl-U** to delete a field, the error message "Do you wish to delete field" might not have been completely erased.

In version 1.1 of dBASE III PLUS, the error message is completely erased.

Sample Programs

In dBASE III PLUS version 1.0, the sample program catalog file, Triplog.CAT, had erroneous entries in the Code field. This caused errors when this Catalog was active and you attempted to select files associated with the database file currently in use.

Now in version 1.1 of dBASE III PLUS, the Code field entries reflect the proper values, so that no errors occur when you select associated files from the Catalog.

INDEX / UNIQUE

In dBASE III PLUS version 1.0, INDEXing with SET UNIQUE ON or INDEX ON UNIQUE worked with small database files (less than 100 records), but failed in one of several ways when INDEXing larger database files.

1. INDEX with the UNIQUE clause seemed to execute properly, but created an index file that was neither unique nor complete. For example, LISTing a file that was INDEXed with SET UNIQUE ON displayed one set of records. SETting UNIQUE OFF and LISTing the same file displayed a different set of records.
2. INDEXing UNIQUE or with SET UNIQUE ON could freeze the computer causing one lost cluster on the disk. This left behind two files, an .NDX file containing 0 bytes and a larger .W44 file. The .W44 file is a temporary file created during the indexing process. Under normal circumstances, it is deleted when INDEX terminates.
3. INDEX UNIQUE with large database files (1,650 records or more) sometimes did not freeze the computer, but continued to create an index file until all the available disk storage was consumed, generating the error message "Disk is full when writing file: <filename>." If this happened, you had to press **Esc** to terminate the INDEX command operation and ERASE the incomplete .NDX, .T44, and .W44 files.

Now in version 1.1 of dBASE III PLUS, you can create unique index files from large database files with either SET UNIQUE ON or the UNIQUE clause to INDEX command without encountering any of these problems.

INDEX ON multiple fields

In dBASE III PLUS version 1.0, INDEXing ON <character field> + STR(<numeric field>) + DTOC(<date field>) TO <filename> could have created an unreliable index file. For example, if the resultant index file was not valid, a REPLACE into the character field did not update the index file.

Now in version 1.1 of dBASE III PLUS, you can create a valid index file using a combination of fields with differing data types converted to character.

LIST TO PRINT / SHEET FEEDER

In dBASE III PLUS version 1.0, LIST TO PRINT did not work properly with a sheet feeder and DOS 3.1. Each sheet fed generated the error message "Printer not ready." This appeared to function correctly if you were running under DOS 2.1.

Now in dBASE III PLUS version 1.1, you can print with a sheet feeder running under DOS 3.1 and not get inappropriate "Printer Not Ready" error messages.

SET PRINTER TO

In dBASE III PLUS version 1.0, SETting PRINTER TO a serial port worked only intermittently. Any attempt to SET PRINTER TO COM1 or SET PRINTER TO COM2 could have returned the error message "Printer not ready" although the printer was on-line and ready to print.

In dBASE III PLUS version 1.1, you can SET PRINT TO COM1 or COM2 and under normal circumstances print without receiving the "Printer not ready" error message.

SET PRINT / SET DEVICE TO PRINT

In dBASE III PLUS version 1.0, if you properly configured a serial printer with MODE and then you SET PRINT ON, a later SET PRINT ON or SET DEVICE TO PRINT command returned the "Printer not ready" error message. To avoid this error condition, you had to SET PRINT OFF first. SET DEVICE TO PRINT followed by SET PRINT ON worked as expected. This problem did not occur with a parallel printer.

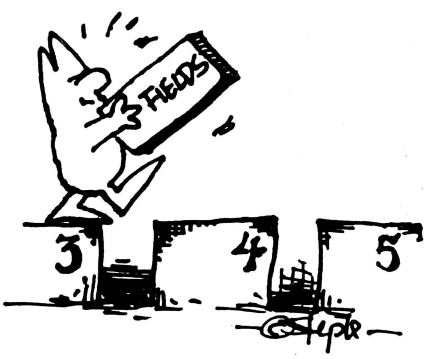
In dBASE III PLUS version 1.1, you can SET PRINT ON and then at some later time SET PRINT ON or SET DEVICE TO PRINT without receiving an error condition if you have configured the printer correctly with MODE.

SORT (ASSIST)

In dBASE III PLUS version 1.0, it was possible to corrupt a database file by SORTing from ASSIST.

Now in dBASE III PLUS version 1.1, you can SORT large database files from ASSIST without corrupting them.

Field Ditto During Data Entry



BY DARYL SEYBOLD

There may be a time when you are APPENDing or EDITing records and you need to CARRY the contents of a field from the same field in the last record. This is known as "dittoing." You can do this at the record level with SET CARRY ON. You may, however, want to this at the field level instead.

The following program, Ditto.PRG, allows you to do this. When you are about to enter a field value and you want the value from the previous record, press **F2** and the field value of the current field is repeated from the previous record. Most, if not all of the functionality of APPEND or EDIT is supported. You can navigate from field to field with **Uparrow** and **Dnarrow** as you would in APPEND or EDIT. **Esc** terminates Ditto and returns control to the dot prompt. Unlike APPEND or EDIT, **Ctrl-End** (**Ctrl-W**) saves the current record and moves control to the next record.

To setup Ditto.PRG, create a command file called Ditto.PRG and enter both Ditto.PRG and the Sayproc procedure. You don't need to create a separate procedure file, Ditto calls itself as a procedure file (yes, in dBASE III PLUS you can do this).

You then call Ditto.PRG passing parameters for the number of fields you want to GET and the name of the database file you want to APPEND into. Note that you are limited to a maximum of 20 fields and only one data entry screen. If you wish more fields or screens, you will have to modify the program to add further capabilities.

To run Ditto.PRG, call it with the following general syntax:

DO Ditto WITH <number of fields>, <.DBF filename>

```
* Program.: Ditto.PRG
* Author..: Daryl Seybold
* Date....: March 1, 1987
* Version.: dBASE III PLUS
* Note(s).: Allows ditto of fields by pressing F2 in an APPEND or
*           EDIT type screen.
*
PARAMETERS numflds, file
*
* ---Define environment.
SET PROCEDURE TO Ditto
SET STATUS OFF
SET TALK OFF
SET SAFETY OFF
SET CONFIRM ON
SET FUNCTION 2 TO CHR(23)
*
* ---Open files.
USE &file
COPY STRUCTURE TO Temp
SELECT B
USE Temp
*
* ---Initialize control variables.
saycol  = 0
getcol  = saycol + 11
more    = .T.
numflds = IIF(numflds > 20, 20, numflds)
*
* ---READKEY() keynames.
ctrlend = 270
uparrow = 4
esc     = 12
*
* ---Main loop.
SET MESSAGE TO "Press F2 for field ditto: Esc to exit"
DO WHILE more
   CLEAR
   APPEND BLANK
   getrow   = 1
   fld_ptr  = 1
   DO Sayproc
  *
  * ---Get current record.
  DO WHILE fld_ptr <= numflds
      field = FIELD(fld_ptr)
      @ getrow, getcol GET &field
      READ
      DO CASE
      CASE MOD(READKEY(),256) = uparrow
      *
      * ---Update pointers, preventing backward movement
      * ---past the first field.
      getrow  = IIF(getrow  = 1, 1, getrow - 1)
         fld_ptr = IIF(fld_ptr = 1, 1, fld_ptr - 1)
```

```
         LOOP
      CASE READKEY() = ctrlend
         *
         * ---Ditto the current field.
         IF RECCOUNT() = 1
         *
         * ---First record in session cannot be dittoed.
            ?? CHR(7)
            SET MESSAGE TO "First entry cannot carry, press any key to continue..."
            @0, 9 SAY SPACE (0)
            WAIT ""
            SET MESSAGE TO "Press F2 for field ditto: Esc to exit"
            getrow  = IIF(getrow  = 1, 1, getrow - 1)
            fld_ptr = IIF(fld_ptr = 1, 1, fld_ptr - 1)
            LOOP
         ELSE
            SKIP - 1
            newvar = &field
            SKIP + 1
            REPLACE &field WITH newvar
            @ getrow, getcol GET &field
            CLEAR GETS
         ENDIF
      CASE READKEY() = esc
         more = .F.
         EXIT
      ENDCASE
      getrow  = getrow + 1
      fld_ptr = fld_ptr + 1
   ENDDO
ENDDO
*
* ---Update master database file from Temp.DBF.
USE
SELECT A
APPEND FROM Temp
USE
ERASE Temp.DBF
*
* ---Close up.
SET MESSAGE TO
SET TALK ON
SET FUNCTION 2 TO "ASSIST;"
CLOSE PROCEDURE
RETURN
* EOP Ditto.PRG
```

```
PROCEDURE Sayproc
* Note(s) ...: Paints the screen like APPEND or EDIT.
*
sayfld_ptr = fld_ptr
sayrow     = getrow
DO WHILE sayfld_ptr <= numflds
```

```
   fld = FIELD(sayfld_ptr)
   @ sayrow, saycol SAY fld
   @ sayrow, getcol GET &fld
   sayfld_ptr = sayfld_ptr + 1
   sayrow     = sayrow + 1
ENDDO
CLEAR GETS
RETURN
* EOP Sayproc
```

Converting Numbers to Words

BY KENNETH N. GETZ AND
KAREN ROBINSON

There have been many dBASE III PLUS programs written to convert numbers into words. This one isn't different, it's just faster. Num2Word.ASM is an assembly language routine that you can LOAD and CALL from dBASE III PLUS to convert numbers into their word equivalents. The most obvious use is for a check writing program, although it is not limited to that. You can use it in any dBASE III PLUS program that requires a numbers-to-words conversion.

Num2Word accepts a number in string format as its parameter and returns the number spelled out in words. For example, using this program, 222.79 becomes TWO HUNDRED TWENTY TWO AND 79/100. The number you convert must be greater than or equal to zero and less than 10 million. To convert a number to a string, use the STR() function with the length parameter as 10 and the decimal parameter as 2, even if the number is smaller than that. For example,

```
STR(x, 10, 2)
```

where x is a numeric memory variable.

When you use Num2Word, you must be aware of two special cases. The first is that numbers less than 1 return a string with "ZERO AND" preceding them. For example, the number .75 is converted to "ZERO AND 75/100." The second is that negative numbers do not convert properly. To ensure that all numbers are positive, use the ABS() function as the first argument of the STR() function:

```
STR(ABS(x), 10, 2)
```

Before CALLing Num2Word, first consecutively initialize the following two memory variables, so they are stored contiguously in memory:

```
number = STR(0, 10, 2)
result = SPACE(254)
```

The memory variable "number" stores the string value of the number. "Result" stores the result of the program; that is, the number spelled out in words. As stated before, the length of "number" must always be 10 with two decimal places.

Passing Parameters

When you pass "number" as a parameter, Num2Word receives the address for "number" from dBASE III PLUS, and from that calculates the address for "result." Once it has the address for "result," Num2Word can store the spelled out version of "number" to "result." For this reason, the memory variables must be defined in the order shown and, once defined, their lengths cannot be changed. For more information regarding this method of passing parameters, refer to "Passing Parameters to Assembly Language Subroutines" in the October 1985 issue of ***TechNotes***, and "Another Wordwrap" in the February 1987 issue of ***TechNotes***.

Setup

Enter the code for Num2Word.ASM using MODIFY COMMAND or any text editor that writes to standard ASCII text files.

To assemble Num2Word.ASM to a .BIN file, the following files must be available, either in the same subdirectory as Num2Word.ASM or through the DOS PATH command: MASM.EXE, LINK.EXE, and EXE2BIN.EXE. Follow the steps below to assemble Num2Word.ASM as a .BIN file.

1. From the DOS prompt type,

   ```
   MASM Num2Word.ASM;
   ```

   This creates an object file (NUM2WORD.OBJ).

2. Then type,

   ```
   LINK NUM2WORD;
   ```

   This creates NUM2WORD.EXE.

3. To create Num2word.BIN type,

   ```
   EXE2BIN NUM2WORD
   ```

4. Last, delete NUM2WORD.EXE,

   ```
   ERASE NUM2WORD.EXE
   ```

Usage

To use the program, LOAD Num2Word into memory, initialize the two memory variables "number" and "result" as shown below:

```
number = STR(x, 10, 2)
result = SPACE(254)
```

where x is a numeric memory variable.

Then CALL Num2Word WITH a number, in string format. For example, if you are converting numbers stored in a database, your program might look something like this:

```
LOAD Num2Word
number = STR(0, 10, 2)
result = SPACE(254)
USE Test
* ---Num is a numeric field.
DO WHILE .NOT. EOF()
    number = STR(ABS(Num), 10, 2)
    CALL Num2Word WITH number
    ? number
    ? TRIM(result)
    ?
    SKIP
ENDDO
CLOSE DATABASE
RETURN
```

```
; Program ...: Num2Word.ASM
; Author ....: Kenneth N. Getz
; Date ......: March 1, 1987
; Version ...: dBASE III PLUS or the Developer's Release
;
MAXLEN  EQU     10              ; Length of longest number name (in English).
LAST    EQU     7               ; Maximum number of digits to worry about.
;-----------------------------------------------------------------------------
SAVE    MACRO   R1,R2,R3,R4,R5,R6,R7,R8
        IRP     X,<R1,R2,R3,R4,R5,R6,R7,R8>
                IFNB    <X>
                        PUSH X
                ENDIF
        ENDM
        ENDM
;-----------------------------------------------------------------------------
RESTORE MACRO   R1,R2,R3,R4,R5,R6,R7,R8
        IRP     X,<R8,R7,R6,R5,R4,R3,R2,R1>
                IFNB    <X>
                        POP X
                ENDIF
        ENDM
        ENDM
;-----------------------------------------------------------------------------
MOVSEG  MACRO   R1,R2
        PUSH    R2
        POP     R1
        ENDM
;-----------------------------------------------------------------------------
GOTOP   MACRO   group
LOCAL   G1,G2
; Get SI to point to beginning of the correct string.  Assumes ES = CS and
; CL = a digit from input string.
;
```

```
        PUSH    DI                      ; Save DI's location.
        MOV     DI,OFFSET CS:&group     ; Point DI to beginning of lookup.
        SUB     CL,'0'                  ; Convert to numeric value.
        AND     AL,0                    ; Put 0 in AL for SCASB.
        CMP     CL,0                    ; See if we got a 0.
        JE      G2                      ; If so, leave DI alone and quit.
G1:     PUSH    CX                      ; Otherwise, save value of digit.
        MOV     CX,MAXLEN               ; Move in maximum string length.
        REPNE   SCASB                   ; Search for a NULL (matches AL).
        POP     CX                      ; Restore the digit value.
        LOOP    G1                      ; Decrement and loop until CX = 0.
G2:     MOV     SI,DI                   ; Point SI to wherever DI is.
        POP     DI                      ; Restore DI's location.
        ENDM
;-------------------------------------------------------------------------------
DASH    MACRO
LOCAL   DONE
; Insert a dash into the output stream if necessary.
;
        PUSH    AX
        CMP     CS:PREPRT,1             ; See if anything's printed yet.
        JE      DONE                    ; If not, no dash possible.
        CMP     BYTE PTR [BX+SI-1],'0'  ; Check to see if previous char was 0.
        JE      DONE                    ; If so, don't need a dash.
        DEC     DI                      ; Otherwise, go back one character.
        MOV     BYTE PTR [DI],'-'       ; Put in the hyphen.
        INC     DI                      ; Put DI back where it was.
DONE:   POP     AX
        ENDM
;-------------------------------------------------------------------------------
COPYNUM MACRO   set
; Copy a number word found in lookup table to output string.
;
        PUSH    SI                      ; Save position counter
        GOTOP   set                     ; Point SI to start of correct string.
        CALL    COPY                    ; Copy the string to output.
        POP     SI                      ; Restore position counter.
        ENDM
;-------------------------------------------------------------------------------
COPYWD  MACRO   wd
; Copy a word from data section to output string.
;
        PUSH    SI
        LEA     SI,CS:&wd               ; Point SI to word in question.
        CALL    COPY                    ; Copy to output.
        POP     SI
        ENDM
;*******************************************************************************
CODE    SEGMENT BYTE PUBLIC     'CODE'
NUM     PROC    FAR
        ASSUME  CS:CODE,ES:CODE
START:  JMP     ENTRY
PREPRT  DB      1                       ; Memvar to indicate whether or not
                                        ; anything's been printed yet.
ONES    DB      'DUMMY',0
        DB      'ONE',0
        DB      'TWO',0
```

```
        DB      'THREE',0
        DB      'FOUR',0
        DB      'FIVE',0
        DB      'SIX',0
        DB      'SEVEN',0
        DB      'EIGHT',0
        DB      'NINE',0
TEENS   DB      'TEN',0
        DB      'ELEVEN',0
        DB      'TWELVE',0
        DB      'THIRTEEN',0
        DB      'FOURTEEN',0
        DB      'FIFTEEN',0
        DB      'SIXTEEN',0
        DB      'SEVENTEEN',0
        DB      'EIGHTEEN',0
        DB      'NINETEEN',0
TENS    DB      'DUMMY',0
        DB      'TEN',0
        DB      'TWENTY',0
        DB      'THIRTY',0
        DB      'FORTY',0
        DB      'FIFTY',0
        DB      'SIXTY',0
        DB      'SEVENTY',0
        DB      'EIGHTY',0
        DB      'NINETY',0
MILLION DB      'MILLION',0
HUNDRED DB      'HUNDRED',0
THOU    DB      'THOUSAND',0
ZERO    DB      'ZERO',0
AND     DB      'AND',0
TAG     DB      '/100',0
ENTRY:  SAVE    AX,BX,CX,DX,DI,SI,DS,BP
        PUSH    DS
        MOV     DI,BX                   ; BX points to first char of input.
        ADD     DI,LAST+5               ; DI now points to outgoing string.
        MOVSEG  ES,DS                   ; Point ES to dBASE's DS.
        CALL    CLEAR                   ; Clear out the output string.
        AND     SI,0                    ; SI = 0 for big loop index.
XLOOP:  AND     CH,0                    ; Clear out high byte.
        MOV     CL,BYTE PTR[BX+SI]      ; Move a single digit into CX.
        CMP     CL,' '                  ; See if we've got a space.
        JNE     TRY1                    ; If not, keep checking.
        JMP     NEXT                    ; If it is a space, go to next digit.
TRY1:   CMP     CL,'0'                  ; See if we got a zero.
        JNE     LOOKUP                  ; If not, keep checking.
        CMP     CS:PREPRT,1             ; See if this is first item found.
        JNZ     MIDDLE                  ; If not, it's not a leading zero.
        COPYWD  ZERO                    ; Otherwise, we write out 'ZERO'.
MIDDLE: JMP     NEXT                    ; Go on to next digit.
LOOKUP:
        MOVSEG  ES,CS                   ; Point ES to our CS.
MILLIONS:
        CMP     SI,0                    ; At first digit?
        JA      HUNDRED_THOUSANDS       ; No -- check hundred-thousands.
        COPYNUM ONES                    ;       Copy number to output.
```

```
        COPYWD  MILLION                 ;       Copy word 'million' to output.
        JMP     NEXT                    ;       Go on to next digit.
HUNDRED_THOUSANDS:                      ;
        CMP     SI,1                    ; At second digit?
        JA      TEN_THOUSANDS           ; No -- check ten-thousands.
        COPYNUM ONES                    ;       Copy number to output.
        COPYWD  HUNDRED                 ;       Copy word 'hundred' to output.
        JMP     NEXT                    ;       Go on to next digit.
TEN_THOUSANDS:
        CMP     SI,2                    ; At third digit?
        JA      THOUSANDS               ; No -- check thousands.
        CMP     CL,'1'                  ; Else: See if we've got a '1'.
        JNE     NOT_1000_TEEN           ;       If not, it's not a teen.
        CMP     BYTE PTR[BX+SI+1],'0'   ;       See if next char is a 0.
        JE      NOT_1000_TEEN           ;       If not, it's not a teen.
        INC     SI                      ; If teen, get next digit.
        AND     CH,0                    ;       Clear out high byte.
        MOV     CL,BYTE PTR[BX+SI]      ;       CX gets single digit.
        COPYNUM TEENS                   ;       Copy out the correct teen.
        COPYWD  THOU                    ;       Copy word 'thousand' to
                                        ;       output.
        JMP     NEXT                    ; Go on to next digit.
NOT_1000_TEEN:                          ; If it's not a teen:
        COPYNUM TENS                    ;       Copy out tens word.
        CMP     BYTE PTR [BX+SI+1],'0'  ;       See if next digit is a '0'.
        JNE     NO_ZERO                 ;       If not, move on.
        COPYWD  THOU                    ;       If so, copy 'thousand' out.
NO_ZERO:                                ; If not a teen, not multiple of
                                        ; 10000.
        JMP     NEXT                    ;       Go on to next digit.
THOUSANDS:
        CMP     SI,3                    ; At fourth digit?
        JA      HUNDREDS                ; No -- check hundreds.
        DASH                            ; Else put in a hyphen if necessary.
        COPYNUM ONES                    ; Copy out the correct units digit.
        COPYWD  THOU                    ; Copy word 'thousand' to output.
        JMP     NEXT                    ; Go on to next digit.
HUNDREDS:
        CMP     SI,4                    ; At fifth digit?
        JA      TENSPOT                 ; No -- check tens.
        COPYNUM ONES                    ; Otherwise copy out correct word.
        COPYWD  HUNDRED                 ; Copy word 'hundred' to output.
        JMP     NEXT                    ; Go on to next digit.
TENSPOT:
        CMP     SI,5                    ; At sixth digit?
        JA      ONESPOT                 ; No -- check ones.
        CMP     CL,'1'                  ; Else see if we got a '1'.
        JNE     NOT_TEEN                ; If not, it's not a teen.
        INC     SI                      ; If teen, get next digit.
        AND     CH,0                    ;       Clear out high byte.
        MOV     CL,BYTE PTR[BX+SI]      ;       CX gets single digit.
        COPYNUM TEENS                   ;       Copy out correct word.
        JMP     NEXT                    ;       Go on to next digit.
NOT_TEEN:                               ; If not a teen:
        COPYNUM TENS                    ;       Copy out correct tens word.
        JMP     NEXT                    ;       Go on to next digit.
ONESPOT:
```

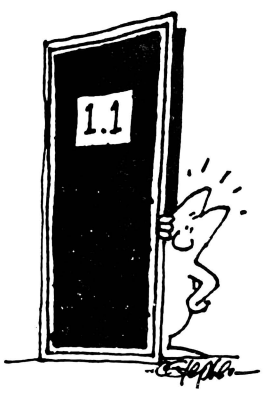
```
        CMP     SI,6                    ; At seventh digit?
        JA      NEXT                    ; No -- we're done.
        DASH                            ; Otherwise, put hyphen if necessary.
        COPYNUM ONES                    ; Copy out correct word.
NEXT:   INC     SI                      ; Go to next position.
        CMP     SI,LAST                 ; See if at last position.
        JAE     FINISH                  ; Yes -- put on finishing touches.
        JMP     XLOOP                   ; Else go back for more.
FINISH: INC     SI                      ; Skip over decimal.
        POP     DS                      ; Get dBASE's DS back.
        CALL    CENTS                   ; Tag on the cents/100.
        OR      CS:PREPRT,1             ; Set up for next call.
        RESTORE AX,BX,CX,DX,DI,SI,DS,BP
        RET
NUM     ENDP
;-------------------------------------------------------------------------
CENTS   PROC    NEAR
; Copy out the cents figure at the end of the string.
;
        MOVSEG  ES,DS                   ; Point ES to dBASE's DS.
        COPYWD  AND                     ; Copy out 'AND'.
        MOV     CX,2                    ; Set counter for two loops.
CLOOP:  MOV     AL,BYTE PTR [BX+SI]     ; Get the next character
        STOSB                           ; Store it at [DI].
        INC     SI                      ; Go to next input character.
        LOOP    CLOOP                   ; Go back for second character.
        LEA     SI,CS:TAG               ; Point SI to final tag.
        CALL    COPY                    ; Copy it out.
        RET
CENTS   ENDP
;-------------------------------------------------------------------------
CLEAR   PROC    NEAR
; Clears out old output string before filling it with new data.
;
        SAVE    AX,CX,DI
        AND     CH,0                     ; Clear out high byte.
        MOV     CL,BYTE PTR [BX+LAST+4] ; Get length of output string.
        DEC     CL                       ; Decrement to skip null byte.
        MOV     AL,' '                   ; Move space to AL for STOSB.
        REP     STOSB                    ; Fill string with spaces.
        RESTORE AX,CX,DI
        RET
CLEAR   ENDP
;-------------------------------------------------------------------------
COPY    PROC    NEAR
; Copies a string from the CS to dBASE's DS.
; Assumes that DI points to output string and SI points to a lookup table.
;
        PUSH    DS                      ; Save dBASE's DS.
        AND     BYTE PTR CS:PREPRT,0    ; Indicate that digit found.
        MOVSEG  ES,DS                   ; ES := DS for DI's use.
        MOV     CX,MAXLEN               ; CX := MAXLEN for string lookup.
LOOP2:  LODS    BYTE PTR CS:[SI]        ; Move a character into AL.
        CMP     AL,0                    ; See if we're at the end of a string.
        JE      L2DONE                  ; If so, jump out.
        STOSB                           ; Else copy AL to where DI points.
        LOOP    LOOP2                   ; Go back for more.
```

```
L2DONE: MOV     AL,' '                  ; Add a space at end.
        STOSB                           ; Store the space.
        POP     DS                      ; Restore dBASE's DS.
        RET
COPY    ENDP
;---------------------------------------------------------------------------
CODE    ENDS
;***************************************************************************
        END     START
```

Num2Word is a routine that you can use with any dBASE III PLUS program that requires numbers-to-words conversion. For more information on using the dBASE III PLUS LOAD and CALL commands, refer to the Commands section in *Learning and Using dBASE III PLUS.*

Acknowledgements: The series of five assembly language programs published in the past four issues of ***TechNotes*** is dedicated to Dr. Paul G. Bamberg and Dr. Henry H. Leitner of Harvard University, for their continuing support, guidance, and friendship over the past two years.

Change Summary for Framework II Version 1.1



BY GAIL KULIKOWSKY

This is a listing of Framework II version 1.0 anomalies corrected in version 1.1. They are listed here so that you can get a clear picture of which anomalies have been corrected and which are still outstanding.

Please note, there is no cross-reference to Framework II anomalies reported in past issues of ***TechNotes.***

.TCM Files

In Framework II version 1.0, if you renamed a .TCM file so that the extension was not .TCM, you could bring the file to the desktop as frame, but any attempt to open or **ZOOM (F9)** the frame would cause your computer to freeze.

In version 1.1, attempting to load a renamed .TCM file results in the error message "Frame badly formatted - unable to read it."

@chr(255)

In Framework II version 1.0, importing a .FW spreadsheet file containing **@chr**(255) caused your computer to freeze.

In version 1.1, all values from **@chr**(0) through **@chr**(255) are imported correctly.

@int

In Framework II version 1.0, **@int** returned incorrect values for numbers that fell in the range 32,768 through 65,535. The value returned was the correct value, minus 65,536.

This is corrected in version 1.1.

@mid

In Framework II version 1.0, if 0 was given as the count parameter for **@mid**, there was a possibility that your computer might freeze.

In version 1.1, Framework II returns the error constant #VALUE! and the message line displays the error message "Parameter must be greater than 0."

@run

In Framework II version 1.0, if a frame formula contained an **@run** and **Print Control Options:Recalc Formula during Printing** was ON, the desktop display would be corrupted or your computer would freeze the next time a **Print** submenu was accessed.

This is corrected in version 1.1.

@tan

In Framework II version 1.0, when **@tan**(0) was recalculated several hundred times and a math coprocessor was in use, it was possible to corrupt memory. This occurred only on systems with a math coprocessor.

This is corrected in version 1.1.

@trace/@int/@abs

In Framework II version 1.0, if **@trace** was set #ON, **@int** and **@abs** returned erroneous values. For example, **@int**(5.0) returned 0 and **@abs**(2.2) returned #N/A!. This did not occur on systems with an arithmetic coprocessor.

In version 1.1, the functions **@int** and **@abs** return correct results with **@trace** #ON or #OFF.

@writetextfile

1. In Framework II version 1.0, if you used a local variable as the framename parameter with **@writetextfile**, your computer would freeze. This problem occurred only under DOS versions 3.0 and above.

   In version 1.1, local variables may be used as the framename parameter within **@writetextfile**.

2. In Framework II version 1.0, if a local variable was used as the pathname parameter with the **@writetextfile** function, the target file was not created, but no error message was displayed.

   In version 1.1, if a local variable is used as the pathname parameter, Framework II returns the message error "Local variable not allowable."

Apps:Print Labels and Mailmerge Using

In Framework II version 1.0, when printing form letters or labels, only the first line of a multiline field in a database was centered or flush right if these options were set ON from the **Words** menu. The second and successive lines of the field were always left-justified.

This is corrected in version 1.1.

Apps:Print Mailmerge Using

In Framework II version 1.0, using **Apps:Mailmerge using** returned the error message 'Unmatched right field marker(" > "); use " > > " to print " > "' when a marker for a single-line field directly followed a marker for a field with more than two lines. The following example would generate the error message.

```
[Database]
Field1                  Field2
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

this is line 1          this is line 1
                        this is line 2          These three lines are
                        this is line 3          contained in the
                                                same database cell.

[Merge text]
< Field2 > < Field1 >
```

If the fields were reversed so Field2 followed Field1, the merge worked correctly.

In version 1.1, the placement of the fields has no effect on the merge.

Apps:Spelling Check

1. In Framework II version 1.0, when a misspelled word was edited and misspelled again, the spelling checker caught the error and prompted you to correct it. However, when the next occurrence of the original misspelled word was encountered, **Auto Replace** gave the second misspelling rather than the correct spelling.

   This is corrected in version 1.1.

2. If Framework II version 1.0 resided on a hard disk but you invoked it from the A> prompt and then sometime later invoked the **Begin Spelling Check** option from the **Apps:Spelling Check** submenu, Framework II looked to the A drive for the SPELL.OVL and dictionary files. The message "Please insert Spelling Check Disk in Drive A: (press ESC to continue)" then displayed, and you had to press **Esc** to continue.

   In version 1.1, Framework II looks for the SPELL.OVL and dictionary files on the drive Framework II resides and loads from.

Apps:Spelling Check and Macros

In Framework II version 1.0, attempting to execute a macro while running the spelling checker could cause your computer to freeze.

In version 1.1, if you press an **Alt-<key>** during a spell check, Framework II beeps and does not act on the macro.

Apps:Spelling Check and Repeated Words

In Framework II version 1.0, the spelling checker ignored all white space (tabs, spaces, end-of-line characters) when checking for repeated words. For example,

```
This is a repeated
      repeated word.
```

In this example, the word "repeated" would be flagged as a repeated word.

In version 1.1, Framework II does not flag a repeated word if it is separated by at least three word delimiters (tabs, spaces, or end-of-line characters).

Apps:Spelling Check:Write Personal Dictionary File

1. If you ran Framework II version 1.0, from a floppy and attempted to write a personal dictionary from the **Apps:Spelling Check** submenu with a word that was exactly 30 characters long, your computer would freeze after Framework II finished writing the dictionary.

   In version 1.1, words from 1 to 30 characters in length can be successfully written to a personal dictionary.

2. In Framework II version 1.0, words between 16 and 30 characters in length with embedded capital letters were not written to a personal dictionary. When Framework II encountered one of these words, it flagged it as a word that was too long. If you selected **Go On,** the dictionary file was written, but it only included words up to the flagged word.

   Version 1.1 of Framework II corrects this problem.

COPYing (F8) Absolute Cell References

In Framework II version 1.0, when **COPYing (F8)** a cell with an absolute reference to a cell in another spreadsheet, the absolute reference was sometimes lost if both column and row references were not absolute. The absolute reference ($) was lost in a vertical copy if the row was not absolute, and it was lost in a horizontal copy if the column was not absolute. For example,

```
[SS1]                [SS2]
   A  B                 A  B
   1  123               1  SS1.$A1
   2  246               2  SS1.A1 <——copiedfromSS2.A1
   3                    3  SS1.A$1 SS1.B1 <——copiedfromSS2.A3
```

In version 1.1, the absolute reference is correctly copied with the cell reference.

COPYing (F8) Boldfaced Text

In Framework II version 1.0, the following steps would cause part of the text in a word frame to be inaccessible, and a **COPY (F8)** operation would behave incorrectly.

```
Ctrl-CE          ; Create empty word frame.
Dnlevel
Ctrl-WB          ; Set bold on.
thisisatest      ; Enter text.
Return
Ctrl-WN          ; Set normal characters on.
thisisatest      ; Enter text.
Ctrl-Home        ; Move to beginning of
                 ; frame.
F6               ; Extend select the bold line.
End
F8               ; Copy it to
Dnarrow          ; the second line.
Return
```

When these steps were followed the first line was copied to the third line instead of the second line, and moving the cursor on the third line would only advance it to the fifth character position. If **Edit:Display Hidden Characters** was ON, the entire text was displayed as graphic characters. If there were spaces in the first line, this sequence still copied the text to the third line, but the other problems did not appear.

This is corrected in version 1.1.

COPYing (F8) Frames

In Framework II version 1.0, **COPY**ing **(F8)** a frame into the beginning of another word frame that already contained text corrupted the contents of the target frame.

In version 1.1, the source frame is copied to the same level as the target frame and the text is not damaged.

COPYing (F8) Soft Hyphens and Inserted Page Breaks to a Spreadsheet

In Framework II version 1.0, sometimes **COPYing (F8)** text from a word frame to a spreadsheet frame displayed garbage characters. This happened if the text you copied contained a word with a soft hyphen, and the hyphenated word was followed by an inserted page break.

In version 1.1, all text is copied correctly.

Ctrl-Del

In Framework II version 1.0, if you used **Ctrl-Del** to delete a one-character word, both that word and the one to its right were deleted.

In version 1.1, **Ctrl-Del** correctly deletes words of any length.

Ctrl-Rightarrow

In Framework II version 1.0, if a one-character word was highlighted and **Ctrl-Rightarrow** was pressed, the word to the right of the one-character word was highlighted.

In version 1.1, pressing **Ctrl-Rightarrow** highlights the one-character word and the blank spaces that follow it.

dBASE File Handles

In Framework II version 1.0, loading dBASE files used file handles that were not returned. If enough dBASE files were loaded, Framework II would eventually return the message "Unable to open file," when an attempt was made to open additional files.

In version 1.1, the number of dBASE files imported is limited by the amount of available memory.

Deleting Soft Hyphens

In Framework II version 1.0, a soft hyphen could not be deleted using **Backspace** if you were at the beginning of the line below a soft hyphen. To remove it you had to position the cursor over the soft hyphen and use **Del** or move the cursor to the end of the line with the soft hyphen and then **Backspace** over it.

This is corrected in version 1.1.

Disk:Import:dBASE II/III

1. In Framework II version 1.0, when a dBASE file was imported, blank numeric fields (no initial value entered) were converted to character fields and contained the formula "0 ".

   In version 1.1, blank numeric fields are imported as numeric 0.

2. In Framework II version 1.0, it was not possible to import dBASE II or dBASE III files with more than 2,304 records. At record 2,304, Framework II dropped back to DOS and displayed the message "Divide overflow."

   In version 1.1, the number of dBASE II or dBASE III records imported is limited only by the amount of available memory.

Disk:Import:WordStar

In Framework II version 1.0, when you imported WordStar files, the **Print Format:Lines per Page** was set to 5, the **Print Format:Print Appearance:Headers:Top Margin** was set to 1, and the **Print Format:Headers:Left** contained the string "here is left header."

In version 1.1, **Print Format:Print Appearance:Lines per Page** is left at the default of 66, the **Print Format:Headers:Top Margin** is set to 5, and there is no default string entered for the **Print Format:Headers:Left.**

DRAGging (F3) Graph Frames

In Framework II version 1.0, you were allowed to **DRAG (F3)** a graph frame off the right side or off the bottom of the desktop.

In version 1.1, attempting to drag a graph off the desktop collapses (resizes) the graph frame.

Edit:Display Hidden Characters

In Framework II version 1.0, if the **Edit:Display Hidden Characters** menu option was turned ON or OFF while inside a frame, only the frame that was selected was initially changed. If the desktop was refreshed, all frames were displayed correctly. Also, entering **HELP (F1)** would turn **Edit:Display Hidden Characters** OFF.

In version 1.1, toggling **Edit:Display Hidden Characters** ON and OFF affects all frames, and **HELP (F1)** does not turn the option OFF.

Edit:Undo

In Framework II version 1.0, executing the **Edit:Undo** option after recalculating **@hide** caused your computer to freeze.

This is corrected in version 1.1.

Edit:Undo and @hide

In Framework II version 1.0, using **Edit:Undo** to restore the value of an empty cell that had been edited froze the computer. For example, cell A1 contained the value 100. The value 300 was entered to replace it and, while still in formula edit, you deleted everything. Subsequently pressing **Return** and then **Edit:Undo** caused your computer to freeze. This happened only when the entire change was deleted.

In version 1.1, **Edit:Undo** returns the cell value before the edit.

Forms View

In Framework II version 1.0, if a database contained 50 or more columns and was in either Forms or dBASE View, scrolling through menus using the **Leftarrow** or **Rightarrow** while in the database was slow.

In version 1.1, menu scrolling speed while in database Forms View and dBASE View is improved.

Graph:Pie Chart

In Framework II version 1.0, the filling or coloring in of pie slices did not occur on any supported plotters.

In version 1.1, the pie is filled in on Hewlett-Packard compatible plotters.

Hard Spaces (Alt-254)

In Framework II version 1.0, FRED had difficulty evaluating hard spaces (ASCII 254) in formulas, cells, or records. Using functions such as **@len** or **@prompt** in conjunction with an **Alt-254** caused your computer to freeze. Putting an **Alt-254** after a number that was followed by a comma in a spreadsheet or database would also cause the same problem.

In version 1.1, a hard space is handled the same as all other characters.

HELP (F1)

In Framework II version 1.0, if **Numlock** was ON, it was turned OFF upon entering **HELP (F1)**.

In version 1.1, the state of **Numlock** is not changed when exiting **HELP (F1)**.

Keyboard Driver for Tandy 1200 HD

In Framework II version 1.0, the ASCII values 0 through 25 were not accessible on a Tandy 1200 HD computer because Tandy uses a slightly different keyboard.

In version 1.1, a new keyboard driver, T1200.KB, is on the Setup disk. This driver allows access to ASCII values 0 through 25.

Library

In Framework II version 1.0, some menu items that were unavailable when only a File Cabinet was selected were available if both the Library and a File Cabinet were selected. For example, it was possible to set **Words:Bold** ON with a File Cabinet selected, but only if the Library Cabinet was also selected. In this example, the contents of the Library Cabinet were boldfaced but the contents of the File Cabinet were unaffected.

In version 1.1, all menu items that are unavailable when a File Cabinet alone is selected are unavailable whether or not the Library Cabinet is also selected.

Library Macros and Ctrl-Numbers

In Framework II version 1.0, using **Ctrl-Numbers** to repeat a Library macro <n> times did not work properly if the macro was created with **Create:Macro**. Instead of executing the entire macro <n> times, it would repeat the first keystroke of that macro <n> - 1 times and the rest of the macro when the repeat keystroke terminated. Macros created in the formula area of a frame worked properly.

In version 1.1., you execute any macro repeatedly with **Ctrl-Numbers** correctly.

Library.FW2

In Framework II version 1.0, if the Library.FW2 file was larger than 163K, Framework II would not load. Framework II would begin to load and display the desktop, but then freeze, often requiring a cold boot. If disk-based extended memory was used, lost clusters resulted.

This is corrected in version 1.1.

Locate:Replace

1. In Framework II version 1.0, if you selected the IBM color graphics adaptor (text mode) as the screen driver in Setup, **Locate:Replace** searched and replaced text inside a graph frame. This left the cursor inside the graph frame, and pressing **Del** several times resulted in garbage characters in the frame.

   In version 1.1, only the frame title of a graph frame is included in the **Locate:Replace** search.

2. In Framework II version 1.0, if you used **Locate:Replace** in a spreadsheet, it would erroneously replace a cell's display value even when the value was generated by the cell's formula. This caused a discrepancy between the display value and the value that was generated the next time a recalculation was performed.

   In version 1.1, **Locate:Replace** does not replace the display value of a cell generated by a formula. Cell values, however, entered by pressing **Spacebar** are replaced.

Locate:Replacing of Attributed Text

In Framework II version 1.0, replacing multiple occurrences of attributed text resulted in a loss of the attribute after the first replace if **End** was pressed to select **Replace All**. For example, if "the" was replaced with "this" in a paragraph, only the first "this" would be boldfaced if **Replace All** was selected at the prompt.

In version 1.1, the source text retains the attributes of the text it replaces for all occurrences.

Locate:Search

1. In Framework II version 1.0, if a string was searched for and not found, the message "NOT FOUND" was displayed and the search string could be edited. However, when Framework II began searching for the new string, it did not display the "SEARCHING" message.

   In version 1.1, the message "SEARCHING" is redisplayed after the search string is edited.

2. In Framework II version 1.0, performing a **Locate:Search** in a database or a spreadsheet with more than 46 characters in a field or cell and terminating the search (pressing any other key, as instructed on screen) would corrupt the field or cell that contained the last occurrence of the string or it would freeze your computer.

   This is corrected in version 1.1.

Locate:Search and Replace

In Framework II version 1.0, individual words in a database field were not highlighted, during a **Locate:Search** or **Locate:Replace**. Instead, the entire field was highlighted.

In version 1.1, individual words are highlighted in Forms View, and in dBASE and Table View the entire field is highlighted.

Locate:Search on Multiple Spreadsheet or Database Frames

In Framework II version 1.0, when multiple spreadsheet or database frames were selected and searched using **Locate:Search** submenu, aborting the search did not always work. If the search string was in several of the frames, Framework II would "Find" the first occurrence of the string in each of the frames before it would stop the search.

In version 1.1, the search is stopped immediately when the exit option is selected.

Numlock and Nonbreaking Spaces

In Framework II version 1.0, nonbreaking spaces (**Ctrl-Space**) could not be issued when **Numlock** was ON. This was particularly a problem on the new IBM Enhanced keyboards, which default to **Numlock** ON.

In version 1.1, a nonbreaking space can be entered with **Numlock** ON or OFF.

Outline View

In Framework II version 1.0, if an outline label was long enough to be truncated by the page number and the label was attributed, the page number assumed the attribute of the label when viewed with **Frames: View Pagination** ON.

In version 1.1, page numbers do not take on the attributes of the frame label.

Power of Ten

In Framework II version 1.0, if a number that was a power of 10 was divided by an integer that produced a repeating fraction and then multiplied by that same divisor, the result did not always yield the initial number. For example, in Framework II version 1.0, the formula (10000/120) * 120 produced 1,000.

In Framework II version 1.1 corrects this problem and yields the correct result of 10,000 for the formula above.

Print:Format/Control

In Framework II version 1.0, illegal values such as decimal numbers and numbers followed by a hyphen could be entered in several of the options in the **Print:Format** and **Print:Control** menus. Although the value was displayed in the option as entered, the message line displayed a value of 0, and moving off the option returned the value to 0. The options that accepted illegal values are **Offset from Left**, **Down from Page Top**, **Up from Page Bottom**, **New Page Number**, and **Lines to Skip**.

In version 1.1, Framework II returns the message "This menu item may only have integer values" when illegal values are entered.

Print:Output Options:Print Frame Labels

In Framework II version 1.0, the names of free-floating spreadsheet frames were printed even if **Print Frame Labels** was OFF. Free-floating frames are subframes that have **Frames:Allow Free Dragging** set ON.

This is corrected in version 1.1.

Printer Driver for NEC 7710

In Framework II version 1.0, the NEC 7710 printer driver did not print more than one consecutive line in bold. When the printer executed a carriage return the boldface code was turned off.

In version 1.1, the NEC 7710 printer driver has been modified and now sends the boldface code at the beginning of each line.

Printing from Virtual Memory

1. In Framework II version 1.0, if a spreadsheet or database frame had overlapping cell labels and the frame used extended memory, printing the frame occasionally caused garbage characters to appear on the printout or your computer to freeze.

   In version 1.1, cell label overlap is handled correctly.

2. In Framework II version 1.0, printing a database in either dBASE View or Forms View was slow.

   In version 1.1, the printing speed of databases is improved.

Referenced Cell Updating

In Framework II version 1.0, when a spreadsheet cell referenced another cell and the value in the first cell was changed to 0, the value in the second cell did not always change. For example, if the value in cell A1 was 4.5 and cell A2 : = A1, A2 equaled 4.5, changing A1 to 0 did not change the number in A2.

This is corrected in version 1.1.

Referencing Databases and Spreadsheets in the Library

In Framework II version 1.0, referencing spreadsheet cells and database fields in the Library cabinet required the full path reference. For example, Library.Ssname.Cellreference.

In version 1.1, the rules are more consistent with the rules for referencing other frames in the Library. To reference a spreadsheet cell or a database field located anywhere in the Library, you need only specify the frame and the cell or field as you would for a spreadsheet or database frame on the desktop.

Referencing Regions

In Framework II version 1.0, region references within FRED functions would cause each cell in the region to take up memory as if it contained information. Functions such as **@next** and **@sum** caused relatively empty spreadsheets to use large amounts of memory.

In version 1.1, this problem is corrected.

Referencing with English Names

1. In Framework II version 1.0, multipart English name references in a spreadsheet or database were not always evaluated correctly. For example, assume that on the desktop there was a word frame labeled Ss that contained a frame labeled One that contained a frame labeled Two that, in turn, contained a frame labeled Three. If the desktop also held a spreadsheet frame that had a cell referenced by One.Two, an attempt to reference the frame called Ss.One.Two.Three from within the spreadsheet would reference the spreadsheet cell One.Two instead.

   This is corrected in version 1.1.

2. In Framework II version 1.0, when using cursor pointing with English name references, cycling through the absolute reference possibilities with the "$" would only cycle between both relative and both absolute states.

   In version 1.1, using the "$" cycles through all four possible absolute reference states, with both coordinate and English name references.

Reformat.FW2

In Framework II version 1.0, if Reformat.FW2 was used to strip carriage returns from justified text and a line did not end with a hard carriage return, the first character of the next line was deleted. Also, if the last line of text did not end with a hard carriage return, the symbol for ASCII value 1 was left at the bottom of the frame, and Framework II beeped continuously until **Ctrl-Break** was pressed.

In version 1.1, justified text is handled correctly.

Saving Files

In Framework II version 1.0, if the drive cabinet of the default drive was open on the desktop, and you attempted to save a file with an eight-character name and an extension of .FW2, your computer would freeze. Framework II saved the file and positioned the highlight on the Library cabinet. Moving the highlight with the arrow keys, however, caused your computer to freeze.

This is corrected in version 1.1.

SIZE (F4)

In Framework II version 1.0, **SIZE**ing **(F4)** a spreadsheet or database column to a width greater than 255 characters caused a video display problem. While scrolling across the columns, the cursor would jump out of the frame and appear by the file cabinets. Scrolling past the wide column would return the cursor to the proper position inside the database.

In version 1.1, the maximum column width is restricted to 255 characters.

ZOOM (F9)

In Framework II version 1.0, if the first field of a database frame was sized to 0 and that field was highlighted, the entire row was highlighted when **ZOOM (F9)** was pressed. If the cursor was moved to another field, only that field was highlighted.

In version 1.1, the selected field is the only field highlighted in full-screen mode.

Importing and Exporting SYLK Files with Framework II

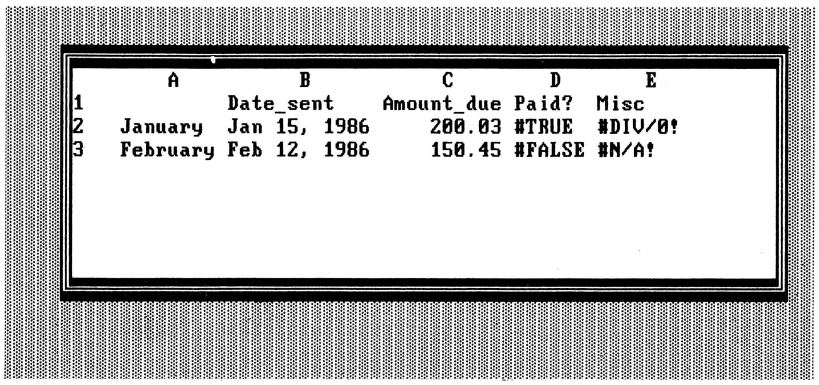
BY QUINN WILDMAN AND KAREN ROBINSON

Although Framework II imports and exports several different file types (MultiMate, dBASE, ASCII, WordStar, and others), it does not handle Microsoft's Multiplan SYLK files. SYLK files are not the format that Multiplan uses directly, but text files that Multiplan can both read and write.

Here we present two FRED programs: Sylkimp.FW2, which imports SYLK files into Framework II spreadsheets, and Sylkexp.FW2, which exports Framework II spreadsheets to SYLK files.

The SYLK File

SYLK is an ASCII text file that has a header section and a data section. The header section includes file structure information such as the number of records and fields; the data section contains the data. SYLK files store two data types: numeric and character. The first record in the data section has the column labels from the Framework II spreadsheet.



	A	B	C	D	E
1		Date_sent	Amount_due	Paid?	Misc
2	January	Jan 15, 1986	200.03	#TRUE	#DIV/0!
3	February	Feb 12, 1986	150.45	#FALSE	#N/A!

Figure 1.

Using the Framework II spreadsheet shown in Figure 1, the SYLK file looks like this:

```
ID;PFW2
F;DG0G10                    Y = number of rows          Header
B;Y3;X5                     X = number of columns
C;Y1;X2;K"Date_sent"
C;X3;K"Amount_due"          Row 1 contains Column labels
C;X4;K"Paid?"
C;X5;K"Misc"
C;Y2;X1;K"January"
C;X2;K"Jan 15, 1986"        Row 2
C;X3;K200.03                                            Data
C;X4;K"#TRUE"
C;X5;K"#DIV/0!"
C;Y3;X1;K"February"
C;X2;K"Feb 12, 1986"        Row 3
C;X3;K150.45
C;X4;K"#FALSE"
C;X5;K"#N/A!"
W;N1;A1 1
E                           End-of-file marker
```

Importing SYLK Files into Framework II

The program, Sylkimp.FW2, imports a Multiplan SYLK file into a Framework II spreadsheet. Because Multiplan has only two data types, all data is imported as either numeric (to five decimal places) or character. Formulas do not transfer, although their results do. Empty cells are ignored. The number of decimal places in numeric data is determined by this command line in Sylkimp.FW2:

```
@performkeys(@decimal(value,5))
```

Change this number to alter the number of decimal places.

The table below outlines how data from a SYLK file is represented in Framework II.

SYLK Data	Framework II Type
Character	Character
Numeric	Numeric (5 decimal places)
Results of Formulas	Numeric
Empty Cells	Ignored

Setup

Sylkimp is written to be used from the **Disk:Import:Import Add-In** menu, although you can use it as a macro. In either situation, first create an empty word frame, label it Sylkimp, enter the program code into the formula area of the frame, and then save it to disk. The SYLK file you import from Multiplan cannot be named Temp and must contain at least two rows and two columns.

To add Sylkimp to the **Disk:Import:Import Add-In** menu, follow these steps:

1. Run the Framework II Setup program.
2. From the first screen, select 2, All Other Uses of the Setup Program.
3. From the Setup Main Menu, select 6, Add-Ins.
4. Select 2, Import Add-Ins.
5. Follow the Setup prompt to enter Sylkimp as the first menu option for the **Disk:Import:Import Add-In** menu (options 1 - 3).

If you want Sylkimp to be option 2 or 3 on the **Disk:Import:Import Add-In** menu, change the following program line to reflect the correct option number:

```
@performkeys("{Ctrl-D}I1{F6}{End}{Leftarrow}{Return}"),
```

change this number to 2 or 3 depending on the option you want.

To execute the import program, select the option from **Disk:Import** menu and, when prompted, enter the filename to import.

To use Sylkimp as a macro, follow these steps:

1. Beginning with line 6, replace the code shown below,

```
; Get filename from import menu.
@performkeys("{Ctrl-D}I1{F6}{End}{Leftarrow}{Return}"),
file := @textselection,
@performkeys("{Esc}{Esc}"),
```

   with

```
; Prompt user for filename
file := @inputline("Enter name of file to import "),
@eraseprompt
```

2. Once this programming change is made, **MOVE (F7)** Sylkimp into the Library, renaming it to,

```
{Alt-I} ;Import a Multiplan SYLK file.
```

To execute the macro, press **Alt-I** and when prompted, enter the filename to import.

```
; Sylkimp.FW2
;
;    by Quinn Wildman
;       March 1, 1987
;
@local(file, rows, columns, frame, currow, curcol, type, flag, value),
@eraseprompt,
@echo(#OFF),
;
; Get file name from import menu.
@performkeys("{Ctrl-D}I1{F6}{End}{Leftarrow}{Return}"),
file := @inputline("Enter the name of the file to be processed."),
@performkeys("{Esc}{Esc}"),
@isabend(#FALSE),
;
; Load the file from disk if it is not accessable on the desktop.
@if(@iserr(@setselection(file)),
   @performkeys("{Ctrl-D}G" & file & "{Return}")
),
@if(@isabend,
   @list(
      @echo(#ON),
      @prompt("File not found."),
      @return(0)
   )
),
@if(@frametype <> 9,
   @list(
      @echo(#ON),
      @prompt("Frame is not a word frame."),
      @return(0)
   )
),
@echo(#ON),
@prompt("Processing - beep indicates one cell processed"),
@echo(#OFF),
frame := @setselection,
@performkeys("{Dnlevel}{Ctrl-Home}{Dnarrow}"),
;
; Line that starts with a B contains coordinates of file.
@while(@textselection <> "B",
   @performkeys("{Dnarrow}")
),
@performkeys("{Ctrl-3}{Rightarrow}{F6}"),
;
; Highlight number of rows.
@while(@mid(@textselection, @len(@textselection), 1) <> ";",
   @performkeys("{Rightarrow}")
),
@performkeys("{Leftarrow}{Return}"),
rows := @textselection,
;
```

```
; SYLK file must have at least 2 rows.
@if(@value(rows) < 2,
   rows := "2"
),
@performkeys("{Ctrl-3}{Rightarrow}{F6}"),
;
; Highlight number of columns; chr(255) indicates end of line.
@while(@mid(@textselection, @len(@textselection), 1) <> @chr(255),
   @performkeys("{Rightarrow}")
),
@performkeys("{Leftarrow}{Return}"),
columns := @textselection,
;
; SYLK file must have at least 2 columns.
@if(@value(columns) <2,
   columns := "2"
),
;
; Create destination spreadsheet.
@performkeys("{Dnarrow}{Home}{Uplevel}{Ctrl-C}H" & rows &
            "{Return}W" & columns & "{Return}STemp{Return}"),
@setselection(frame),
@performkeys("{Dnlevel}"),
@isabend(#FALSE),
@while(@not(@isabend),
   ;
   ; If a line doesn't start with a C skip it.
   @while(@and(@not(@isabend), @textselection <> "C"),
      @performkeys("{Dnarrow}")
   ),
   ;
   ; You are either at the end or have a row with a value for spreadsheet.
   @if(@isabend,
      @list(
         ;
         ; Name output frame the same as original frame.
         @performkeys("{Uplevel}{Del}"),
         @setselection("Temp"),
         @performkeys(frame & "{Return}"),
         @echo(#ON),
         @prompt("Import Completed Successfully."),
         @return(0)
         )
   ),
   flag := #TRUE,
   ;
   ; The only way to get out of the loop is if a value is found.
   @while(flag,
      @while(@and(@textselection <> "X",
                  @textselection <> "Y",
                  @textselection <> "K"
            ),
         @performkeys("{Rightarrow}")
      ),
      ;
      ; If X or Y get new coordinate.
      @if(@or(@textselection = "X", @textselection = "Y"),
```

```
    @list(
       type := @textselection,
       @performkeys("{Rightarrow}{F6}"),
       @while(@mid(@textselection, @len(@textselection), 1) <> ";",
          @performkeys("{Rightarrow}")
       ),
       @performkeys("{Leftarrow}{Return}"),
       @if(type = "X",
          curcol := @textselection,
          currow := @textselection
       )
    ),
    ;
    ; Must be a new value.
    @list(
       ; First check if numeric, character, or error constant.
       @performkeys("{Rightarrow}"),
       @if(@textselection = """",
          ;
          ; Must be a character.
          @list(

               @performkeys("{Rightarrow}{F6}{End}{Leftarrow}"&
               "{Leftarrow}{Return}"),
            value := @textselection,
          ),
          @list(
            ;
            ; Check for error constant.
             @if(@textselection = "#",
                @list(
                   @performkeys("{F6}{End}{Leftarrow}{Return}"),
                   value := @textselection,
                ),
                ;
                ; Must be a numeric.
                @list(
                   @performkeys("{F6}{End}{Leftarrow}{Return}"),
                   value := @value(@textselection)
                )
             )
          )
       ),
       @setselection("Temp"),
       @performkeys("{Ctrl-L}G" & @chr(64 + @value(curcol)) & currow &
                    "{Return}"),
       @if(@isalpha(value),
          @performkeys(" " & value),
          @performkeys(@decimal(value,5))
       ),
       @performkeys("{Return}"),
       @beep(200,20),
       @setselection(frame),
       @performkeys("{Dnlevel}{Dnarrow}{Home}"),
       flag := #FALSE
    )
 )
```

```
   )
),
;
; Now name output frame the same as original frame.
@performkeys("{Uplevel}{Del}"),
@setselection("Temp"),
@performkeys(frame & "{Return}"),
@echo(#ON),
@prompt("Import Completed Successfully.")
```

Exporting From Framework II to SYLK

Sylkexp.FW2 exports data from a Framework II spreadsheet to a Multiplan SYLK file format by first writing the data to a word frame and then writing the word frame to a text file. If you have a database that you wish to export, first copy the data to a spreadsheet.

All Framework II data types are exported as either character or numeric. There is no fixed number of decimals; numeric data is exported with the number of decimals it has. Formulas do not transfer, although their results do. As in the import program, empty cells are ignored.

The table below outlines how Framework II data types are stored in a Multiplan SYLK file.

Fw Data Type	SYLK data type
Character	Character
Numeric	Numeric
Decimals	Variable length
Date	Character
Logical	Character
#TRUE	"#TRUE"
#FALSE	"#FALSE"
Error Constants	Character
#VALUE	"#VALUE"
#REF	"#REF"
#DIV/0	"#DIV/0"
#NUM	"#NUM"
#TBD	"#TBD"
#NAME?	"#NAME?"
#NA!	"#NA!"
Results of Formulas	Numeric
An empty cell	Ignored

Setup

You can add Sylkexp to the **Disk:Export:Export Add-In** menu or use it as a macro. In either case, first enter the program code into the formula area of an empty word frame named Sylkexp, and then follow the instructions below to set up Sylkexp as either an Add-In menu option or as a macro.

To add Sylkexp to the **Disk:Export:Export Add-In** menu, follow these steps:

1. Run the Framework II Setup program.
2. From the first screen, select 2, All Other Uses of the Setup Program.
3. From the Setup Main Menu, select 6, Add-Ins.
4. Select 3, Export Add-Ins.
5. Follow the screen prompts to enter Sylkexp as an option in the **Disk:Export:Export Add-In** menu.

To export data to a Multiplan SYLK file, highlight the Framework II spreadsheet and select the export option from the **Disk:Export** menu.

To set up Sylkexp as a macro, **MOVE (F7)** Sylkexp into the Library, renaming it to:

{Alt-E} ;Export to a Multiplan SYLK file.

To execute the macro, highlight the Framework II spreadsheet and press **Alt-E**.

```
; Sylkexp.FW2
;
;    by Quinn Wildman
;       March 1, 1987
;
@local(frame, name, contents, column, columns, row, rows, newrow, newline),
;
; Store a {Return} in newline.
newline := "
",
;
; Request a spreadheet name.
name := @inputline("Enter the name of the spreadsheet to be processed."),
;
; Test to see that the current frame exists and is a spreadsheet.
@eraseprompt,
@if(@iserr(@setselection(name)) <> #N/A!,
   @if(@frametype <> 6,
      @list(
         @prompt("Frame does not exist or is not a spreadsheet."),
         @return(0)
      )
   )
),
;
; Request the name of the output file.
@eraseprompt,
name := @inputline("Enter the name of the output file."),
;
; Define a default name for the output file.
@if(name = "",
  name := "Sylkout"
),
;
@eraseprompt,
@prompt("Processing - beep indicates one record processed."),
```

```
@echo(#OFF),
frame := @setselection,
@performkeys("{Dnlevel}"),
;
; Put id in first line.
contents := "ID;PFW2" & newline,
;
; Put format in second line.
contents := contents & "F;DGOG10" & newline,
;
; Go to the last cell in the spreadsheet.
@performkeys("{Ctrl-End}"),
;
; Put in size of spreadsheet using panel1 and panel2.
contents := contents & "B;Y" & @integer(@panel2) & ";X" & @integer(@panel1) &
newline,
columns  := @panel1,
rows     := @panel2,
@performkeys("{Ctrl-Home}"),
@isabend(#FALSE),
@while(@not(@isabend),
   newrow := #TRUE,
   @isabend(#FALSE),
   @while(@not(@isabend),
      @if(@not(@isna(@textselection)),
         @list(
            ; C for constant data type first goes before all data.
            contents := contents & "C;",
            ; If first entry for row, enter Y coordinate.
            @if(newrow,
               @list(
                  contents := contents & "Y" & @integer(@panel2) & ";",
                  newrow   := #FALSE
               )
            ),
            ;
            ; Enter X coordinate using panel1.
            contents := contents & "X" & @integer(@panel1) & ";",
            ;
            ; Put data in SYLK file.

            @if(@and(@not(@iserr(@value(@textselection))),
               @mid(@textselection,1,1) <> "#"),
               ;
               ; Data is numeric.
               contents := contents & "K" & @textselection & newline,
               ;
               ; Data is character.
               contents := contents & "K""" & @textselection & """" & newline
            )
         )
      ),
      @performkeys("{Rightarrow}")
   ),
   @prompt("Processing - beep indicates one record processed."),
   @beep(200, 20),
   @isabend(#FALSE),
```

```
   @performkeys("{Dnarrow}{Home}")
),
;
; Write the termination sequence and clean up.
contents := contents & "W;N1;A1 1" & newline & "E",
@performkeys("{Ctrl-Uplevel}{Return}{Ctrl-C}EOutput{Return}{Return}"),
Output := contents,
@setselection(frame),
@echo(#ON),
;
; Write the temp frame to output file.
@prompt("Writing Output File - One Moment Please."),
@writetextfile(name, Output),
@setselection("Output"),
@performkeys("{Del}"),
@eraseprompt,
@prompt("Export Completed Successfully."),
@setselection(frame)
```

For more information regarding the SYLK file format, refer to the Multiplan reference manual.

Creating Endnotes

BY MERYL HORWITZ

MultiMate Advantage's footnoting feature helps organize references in reports, books, and scholarly materials. With it, you can have multiple footnotes per page with up to eight lines per footnote. You can also use the footnoting feature to accomplish other tasks that are not directly supported in MultiMate Advantage, such as printing endnotes instead of footnotes. Endnotes are footnotes that print at the end of the document instead of on the page that references it.

By using MultiMate's file conversion utility, UTIL, you can convert a footnoted document into a document with endnotes. The process works like this: when you export a footnoted document to an ASCII or WPS file format, all the footnotes are written to the last page of the document. So, when you convert it back to a MultiMate document, you can print the document with endnotes instead of footnotes. WPS is a file format defined by Wang Laboratories.

When executing this type of file conversion, be sure to use a different name for the destination filename so that MultiMate does not overwrite the original document (in this case Footnotes.DOC) when it converts back to a MultiMate document.

Using the WPS Format to Create Endnotes

WPS conversion is included with MultiMate Advantage version 3.60. If you have MultiMate Advantage version 3.30 or 3.50, you can contact MultiMate Technical Support for this conversion file.

If you decide to create endnotes using the WPS file conversion, be aware that only the first two headers and footers are retained and further, that they are retained as text. So, when you convert the WPS file back to a MultiMate document, you need to re-enter the header codes.

If you are not familiar with doing file conversion using the UTIL program, follow these steps:

1. Create a document that contains footnotes and name it Footnotes.DOC. For information on how to create footnotes, refer to the *MultiMate Reference Manual.*
2. To move directly to the File Conversion screen in UTIL, type

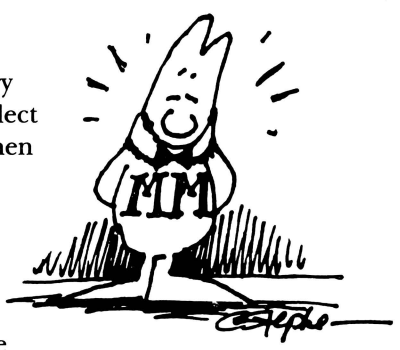
   UTIL F C

   from the DOS prompt.
3. On the File Conversion screen, enter Footnotes.DOC for the Source filename, MM for the file type, and the DOS pathname. Enter Endnotes.WPS for the destination filename, WPS for the file type, and the DOS pathname. If necessary enter the DOS pathname for the location of the UTIL files.
4. Press **F10** to begin conversion.

   When the conversion is complete, convert it back to a MultiMate document file. The steps are listed below:
5. From the File Conversion screen, enter Endnotes.WPS as the Source filename and WPS as the file type. Then, enter Final.DOC as the Destination filename and MM as the type.
6. Press **F10** to convert it back to a MultiMate document file.

When you load this new MultiMate document (Final.DOC), all of the footnotes are listed on the last page of the document. If you wish, you can go through the document and superscript the reference numbers cited within the text and on the endnotes page.

In addition, the Document Summary screen for Final.DOC is titled Footnotes instead of Final. This is because the WPS format retains the Document Summary screen of the original document during file conversion. To change the name, select **MultiMate Document Handling Utilities** (option 6) from the main menu and then **Rename a document** (option 4).



Using the WPS Format to Create ASCII

All versions of MultiMate Advantage have the ASCII file conversion. As with the WPS file conversion, if you had headers in your original document, you have to re-enter them once the file is converted back to a MultiMate document. The text of the headers will appear but the MultiMate header codes do not. The conversion process is very similar to that done with WPS files. The exceptions are listed below:

1. In step 3, enter ASC for the destination file extension and ASCII for the file type.
2. In step 5, enter ASC for the source file extension and ASCII for the file type.

When you convert the file back to MultiMate document, the resultant document will also have an extra page of endnotes. The endnote page is titled "FOOTNOTES" and the reference numbers throughout the document are bracketed. Delete the FOOTNOTES title and the brackets and if you wish, superscript the numbers.

Although MultiMate does not have an endnotes function, you can use MultiMate footnoting capabilities to create endnotes. For more information on how to expand the use of footnotes, refer to *MultiMate Advantage Specific Samples Applications.*

USAGE TIPS

dBASE III PLUS

BY KENNETH N. GETZ

Converting Numbers to Scientific Notation

Although dBASE III PLUS allows you to convert numbers from scientific notation to numbers in decimal notation, it does not allow you convert decimal notation to scientific notation. The following programs, Sci.PRG, converts numbers in decimal format to scientific notation.

To use Sci.PRG, call it with the following general syntax:

```
ret_val = ""
DO Sci WITH <number>, <decimals>, ret_val
```

where the arguments are defined as follows:

```
number  ::= integer to be converted
decimals ::= number of decimal places
ret_val  ::= character variable to accept return value
```

Also included is a sample program, SciTest.PRG, that demonstrates how to use Sci.PRG to list a set of numbers from a database file in scientific notation.

```
* Program ...: Sci.PRG
* Author ....: Kenneth N. Getz
* Date .......: March 1, 1987
* Version ...: dBASE III PLUS
* Note(s) ...: Converts x to scientific notation correct to decimal number
*              of decimal places.  Works for any positive number within
*              dBASE III PLUS limits.
*
PARAMETERS x, dec, ret_val
PRIVATE ALL
y = LOG(ABS(x)) / LOG(10)          && Find order of magnitude.
y = IIF(INT(y) = y, INT(y) + 1, INT(y))
y = IIF(ABS(x) < 1,y - 1,y)
z = x / 10^y                       && Normalize the number (0 < z < 10).
order = 10^dec                     && Find order of rounding accuracy.
fudge = IIF(x < 0, -.5, .5)
q = INT( z * order + fudge)/order
ret_val = LTRIM(STR(q,15,dec)) + "E" + IIF(ABS(x) < 1,"-","+") +;
          LTRIM(STR(ABS(y)))
RETURN
* EOP Sci.PRG
```

```
* Program ...: SciTest.PRG
* Author ....: Kenneth N. Getz
* Date ......: March 1, 1987
* Version ...: dBASE III PLUS
* Note(s) ...: A test program for Sci.PRG.  Uses the database SciTest.DBF
*
PARAMETERS decimals
PRIVATE ALL
STORE SPACE(30) TO ret_val
USE Scitest
DO WHILE .NOT. EOF()
  ? num
  DO Sci WITH num, decimals, ret_val
  ?? IIF(num < 0, SPACE(10), SPACE(11)) + ret_val
  SKIP
ENDDO
RETURN
* EOP SciTest.PRG
```

COPY STRUCTURE EXTENDED/SET CATALOG TO

There are occasions where you want to have dBASE III PLUS CREATE a database file without the presence of another database file in the environment. To CREATE a database file structure from scratch under program control requires the use of COPY STRUCTURE EXTENDED and CREATE FROM. COPY STRUCTURE EXTENDED allows you to create a database file that has as its fields the attributes of a database field. So, each record corresponds to a field in the source database file. And since the structure of the original is held in database file, you can manipulate its definition as you would any other with APPEND, BROWSE, EDIT, LIST, LABEL, and REPORT FORM. To CREATE the new database structure FROM the EXTENDED database file, you use CREATE FROM <extended database file>.

To COPY STRUCTURE EXTENDED, however, requires that you have a database file in USE. In dBASE III, this meant that you had to have prior knowledge of a database file on disk that you could USE before you could create the EXTENDED STRUCTURE database file. dBASE III PLUS unlike its predecessor can create a database file without another database file being present. This database file is the Catalog, the only database file that springs directly from dBASE itself. By SETting CATALOG TO a filename, COPYing the STRUCTURE EXTENDED to a temporary EXTENDED database file, and then ZAPping the temporary EXTENDED file, you can add records to the new database file each record corresponding to a field in the target database file. You can then CREATE the final database file FROM the new file and there is now a fully operational database file where none existed before. The basic method is as follows:

```
SET CATALOG TO <temporary Catalog file>
SELECT 10
COPY STRUCTURE EXTENDED TO <temporary EXTENDED file>
USE <temporary EXTENDED file>
* ---Clear Catalog field definitions.
ZAP
* ---Fill the temporary structure file with field definitions.
APPEND BLANK
REPLACE Field_name WITH <"field name">,;
        Field_type WITH <"field type (C, D, N, or M)">,;
        Field_len  WITH <field length>,;
        Field_dec  WITH <number of decimal places>
USE
* ---Make new database file from temporary structure file.
CREATE <new filename> FROM <temporary EXTENDED file>
```

The following takes this method and demonstrates a complete program for building a database file at runtime. It is made of two command files, Make.PRG which CREATEs the new database structure and Fill.PRG, a data dependent routine that creates the new field definitions. For your usage, create a new version a Fill.PRG for each of the database files that you want to build under program control. To call Make.PRG, use the following general syntax:

```
DO Make WITH <temporary database filename>,;
             <new database filename>
```

```
* Program ...: Make.PRG
* Author ....: Kenneth N. Getz
* Date .....: March 1, 1987
* Version ...: dBASE III PLUS
* Note(s) ...: Creates a database file from scratch.
*
PARAMETERS tempfile, filename
PRIVATE ALL
*
* ---Set up environment.
SET SAFETY OFF
SET TITLE OFF
SET TALK OFF
*
* ---Create the temporary structure file.
SET CATALOG TO &tempfile.
SELECT 10
COPY STRUCTURE EXTENDED TO &tempfile.
USE &tempfile.
ZAP
*
* ---Fill the temporary structure file with field definitions.
DO Fill
CLOSE DATABASES
*
* ---Make new datafile from temporary structure file.
CREATE &filename FROM &tempfile  && Create actual datafile from the structure.
ERASE &tempfile..DBF
ERASE &tempfile..CAT
*
```

```
* ---Restore environment.
SET SAFETY ON
SET TITLE ON
SET TALK ON
RETURN
* EOP Make.PRG
```

```
* Program ...: Fill.PRG
* Author ....: Kenneth N. Getz
* Date ......: March 1, 1987
* Version ...: dBASE III PLUS
* Note(s) ...: Procedure to fill in the structure of a STRUCTURE EXTENDED
*              data file.
*
* ---Create field definitions.
APPEND BLANK
REPLACE Field_name WITH "Name",;
        Field_type WITH "C",;
        Field_len  WITH 25
APPEND BLANK
REPLACE Field_name WITH "Address",;
        Field_type WITH "C",;
        Field_len  WITH 30
APPEND BLANK
REPLACE Field_name WITH "City",;
        Field_type WITH "C",;
        Field_len  WITH 20,
APPEND BLANK
REPLACE Field_name WITH "State",;
        Field_type WITH "C",;
        Field_len  WITH 2
APPEND BLANK
REPLACE Field_name WITH "Zip",;
        Field_type WITH "C",;
        Field_len  WITH 5
APPEND BLANK
REPLACE Field_name WITH "Income",;
        Field_type WITH "N",;
        Field_len  WITH 8,;
        Field_dec  WITH 2
RETURN
* EOP Fill.PRG
```

---

dBASE TOOLS FOR C—The Graphics Library

BY DOUG HAMAN

Array Save and Restore

The Graphics Library uses a binary save of an array, unlike The Programmer's Library, which uses a text save. The *dBASE Tools for C* Incentive Disk allows you to use a text save with The Graphics Library.

Linking the Graphics Library

There have been several requests either to include portions of the Graphics Library in the Programmer's Library at link time or exclude portions of the Graphics Library that require source code. Neither one is possible but you can load both libraries at once as long as you have 640K of memory.

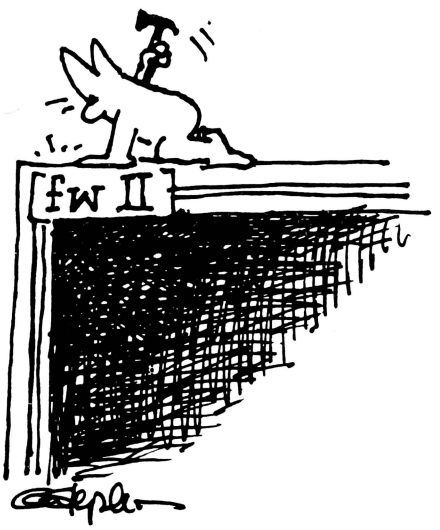
Scaling

The Graphics Library does not currently include parameters to allow manual Y axis scaling (as in Framework II). You can, however, write a dBASE III PLUS using the low-level graphic functions to draw a graph.

Framework II

BY CHRISTOPHER WHITE AND KENT IRWIN

Auto Indent Macro



Most programmers spend more time indenting their code than they do typing it. The following macro is designed to aid the programmer using Framework II as a code editor. Alt-I will toggle auto-indent on and off. When auto-indent is on, the **Tab** adds to the **Words:Left Margin** setting, and **Backtab (Shift-Tab)** subtracts from the **Words:Left Margin** setting restoring the last tab stop. This way, once you indent, all the lines you subsequently type will be at the same level of indentation.

To set up the macro, create a containing frame labelled {Alt-I}. Then create three sub-frames labelled In, Out, State, and Indent. Next, type in the following programs into the formula areas of the indicated frames. Before executing {Alt-I} for the first time, initialize {Alt-I}.State to #FALSE and {Alt-T}.Indent to the number of spaces you want to indent for each tab stop. Do not assign this frame a value less than two. Finally, **MOVE (F7)** the {Alt-I} frame into the Library cabinet and save with **Ctrl-Return**. As you work using {Alt-I}, you can avoid screen flicker when you press **Tab** or **Backtab** by always editing in **ZOOM (F9)**.

```
; {Alt-I}
;
;    by Kent Irwin
;       March 1, 1987
;
@if([{Alt-I}].State,
   @list(
      [{Alt-I}].state := #FALSE,
      @keyfilter({Tab}),
      @keyfilter({Backtab}),
      @eraseprompt,
      @prompt("Auto-Indent is OFF",31)
   ),
   @list(
```

```
      [{Alt-I}].State := #TRUE,
      @keyfilter({Tab}, [{Alt-I}].[In]),
      @keyfilter({Backtab}, [{Alt-I}].[Out]),
      @eraseprompt,
      @prompt("Auto-Indent is ON",31)
   )
)
```

```
; [{Alt-I}].In
;
@echo(#OFF),
@local(margin),
@performkeys("{Ctrl-W}L{F6}{End}"),
margin := @value(@textselection) + indent,
@performkeys("{Del}" & @integer(margin) & "{Return}"),
@echo(#ON)
```

```
; [{Alt-I}].Out
;
@echo(#OFF),
@local(margin),
@performkeys("{Ctrl-W}L{F6}{End}"),
margin := @value(@textselection),
@if(margin > indent - 1,
   @performkeys("{Del}" & @integer(margin - indent) & "{Return}"),
   @performkeys("{Return}{Return}")
),
@echo(#ON)
```

Using Attributes in Print Formulas

One thing you might find difficult to do in Framework II is to set an attribute such as bold, underline, or italic to string that is a result of a calculation. This is particularly important for print formulas where you may want an attribute for a page number or a formatted date. The following user-defined function, Attrib.FW2 allows you to assign attributes to strings including the results of formulas. The general syntax is as follows:

@Attrib(" < string > ", < attribute >)

The algorithm to accomplish this is based on **@chr** which allows as one of its arguments the attribute of the character returned. This attribute argument is a numeric expression between zero and seven. The table below summarizes the combinations of attributes you can assign.

Number	Attribute
0	Normal
1	Bold
2	Italic
3	Bold/Italic
4	Underline
5	Bold/Underline
6	Underline/Italic
7	Bold/Underline/Italic

Attrib merely parses the string you pass into a character stream, uses **@chr** to assign the attribute you want a character at a time, and returns the attributed string when there are no characters left in the stream. You can use this attributed string as the argument for any other formula you want. This includes spreadsheet cells, database fields, or print formulas.

As an example, a typical application of a calculated print formula might include today's date underlined as header left and the page number bolded as footer center. The following block of code demonstrates a print formula that accomplishes this.

```
; Underline date.
@hl(@Attrib(@date4(@today), 4)),
;
; Bold page number.
@fc("- " & @Attrib(@pn, 1) & " -")
```

```
; Attrib.FW2
    ;
    ;  by Christopher White
    ;     March 1, 1987
    ;
    @local(instr, attribute, ptr, outstr),
    instr     := @item1,
    attribute := @item2,
    ptr       := 1,
    outstr    := "",
    @while(ptr <= @len(instr),
       outstr := outstr & @chr(@int(@mid(instr, ptr, 1)), attribute),
        ptr   := ptr + 1
    ),
    @return(outstr & @chr(0, 0))
```

MultiMate

BY MERYL HORWITZ

Merge Printing

When merge printing, if data from the Secondary document does not merge with the text in the Primary document, check for the following conditions:

1. Each record in the Secondary document must end with a **Return** and a page break (**F2**). For example,

   |Name| **Return**
   Larry Jones| **Return**
   |Address| **Return**
   23 Holland Street| **Return**
   |City| **Return**
   Hartford| **Return**
   |State| **Return**
   CT| **Return**
   |Zip| **Return**
   06103| **Return**
   Return

   < **F2** page break here >

   |Name| **Return**
   Jane Robins| **Return**
   |Address|**Return**
   14 Terrace Lane| **Return**
   |City|**Return**
   Seattle| **Return**
   |State| **Return**
   WA| **Return**
   |Zip| **Return**
   22222|**Return**
   Return

2. The merge variable names in the Primary document and the Secondary document must be spelled and capitalized exactly the same. In the example above, the merge variable names are "Name," "Address," "City," etc. The data items are the actual names and address, such as "Jane Robins" and "Seattle."

3. The MultiMate merge code symbol **Alt-M** (|) must precede and follow each merge variable name in both the Primary document and Secondary document.

4. In the Secondary document, the merge code symbol **Alt-M** (|) must follow each data item. For example,

   Jane Robins| **Return**

5. The Secondary document can contain up to 64 merge items per record.

Printing Address Labels

Using the MultiMate Advantage Merge Print Utility, you can print mailing labels on envelopes or address labels by merging the data in your Secondary Document with a Primary Document that is setup in a label format. For example, you can set up your Primary Document like this:

|Name|**Return**
|Address|**Return**
|City|, |St| |Zip|**Return**

And set the values for Left Margin, Top Margin, and Doc. Page Length to the accommodate the size of the envelope or label. For example,

Left Margin = 35
Top Margin = 12
Doc. Page Length = 24 (size of envelope)

Using Tab in a Primary Merge Document

If you use **Tab** to indent or center the label format in your Primary Document, be sure to insert **Tab**s on the second and succeeding lines of data in your Secondary Document; otherwise, only the first line of data is **Tab**bed with the remaining lines printing at the left margin. An example of how to set up your Primary and Secondary Documents is shown below:

Primary Document:

|1.....**Tab**.....**Return**

Tab|Name|**Return**

Secondary Document:

|1.....**Return**
|Name|**Return**
Liz Robinson**Return**
Tab52 Oakland Ave.**Return**
TabTorrance, CA 90502|**Return**
Return

ON-LINE SERVICES

BY STEVE SILVERWOOD

Do you have a modem? Do you subscribe to CompuServe or The Source? If so, Ashton-Tate offers technical support on all of these electronic information services. You can quickly access the kind of quality technical information that you have come to expect from ***TechNotes***, and you can do it through your computer system. To take advantage of this progressive new support method, you must subscribe to one of the services listed below. There are no additional charges for using our on-line support services, apart from the normal connect-time fees for each service. You can get specific subscription information by calling one of the following numbers:

Service	Subscriber Information	Command to find Ashton-Tate
CompuServe	(800) 848-8199	GO ASHTON
The Source	(800) 336-3366	ATSIG

To give you an idea of support that is happening on-line, ***TechNotes*** will be publishing regular excerpts from the Ashton-Tate Forum on CompuServe. This monthly material comes from messages our Forum users have posted on the bulletin board. All message headers have been removed to protect the identities of our users. Every other message within a single conversation topic has been italicized to enhance readability.

This month all messages come from the ATSIG forum on The Source.

Framework II

Who has a good FRED program to accept a month name as a three-character string and return the month number as a two-character string? For example, I want to convert the month string "Nov" to "11." Thanks in advance.

The best way I can think of to return a number from a 3-digit character string is as follows (maybe somebody else has a better idea). I am assuming you are passing a parameter which contains the 3 digit month string and you want to return the number of the month as a character string.

```
; MontoNum.FW2
;
; Called by @MontoNum("<three-digit month name>")
;
@local(month),
@if(@item1 = "Jan", month := "01"),
@if(@item1 = "Feb", month := "02"),
@if(@item1 = "Mar",month := "03"),
@if(@item1 = "Apr", month := "04"),
@if(@item1 = "May",month := "05"),
@if(@item1 = "Jun", month := "06"),
@if(@item1 = "Jul", month := "07"),
@if(@item1 = "Aug",month := "08"),
@if(@item1 = "Sep", month := "09"),
@if(@item1 = "Oct", month := "10"),
@if(@item1 = "Nov",month := "11"),
@if(@item1 = "Dec", month := "12"),
@return(month)
```

.....

Our 60 man law firm is a long-term devoted user of both dBASE III PLUS and Framework II, but we have certainly run into some problems in the database area with Framework applications. I am frustrated trying to write FRED programs for use with Framework II databases. Perhaps my legal background and absence of technical training is an overwhelming obstacle, but I do very well with dBASE III PLUS. Somehow the FRED language and the seeming limitations of Framework II generally always seem to get in my way.

Please let me have your comments on any or all of the following:

1. How does FRED code accomplish the equivalent of APPENDing record in dBASE III PLUS and replacing named fields with already gathered variable data?
2. Why do the experts who write the books always seem to add the new record at the top of the file, not the bottom?
3. Is there any FRED equivalent to the dBASE III PLUS REPLACE ALL FOR <condition> command?

The way to APPEND a record is first either go to a blank record, or add one new record with ***Create:Rows/Records:Add.*** *Then get the information you want input into memory variables using* ***@inputline****, then* ***@performkeys*** *the memory variables. Dataent1.ART, which is the SIG library of files, has an example of how to do this. There another file that goes along with it called Add.FW2, which is in the SIG library of files also.*

In response to your second question, I see no reason for putting a new record at the beginning of the file versus putting it at the end.

As to your third question, you can REPLACE ALL by going to the database field name and altering the field formula to be in the form:

<field name> = <expression>

and the result of the REPLACE expression will be put into every record in the database. For simple replacements, you can use the ***Locate:Replace*** *menu option by highlighting the database field name and executing the* ***Locate:Replace****.*

.....

I am having trouble returning a value from **@panel** using string conversion functions. I have a program with a central database frame that filters the database, and then copies the filtered records to a number of smaller database frames. I want to be able to have the program find out how much room is in each of the small databases before the copy process from the central database occurs. Then, I can automatically add rows when the program finds the receiving database to small for the copy.

I don't think you need to convert panel to a string, but here's how to it, if you need to do so:

memvar : = @integer(@panel1)

The most important thing to remember when you are checking the ***@panel*** *is that you are inside the database frame, otherwise* ***@panel*** *returns information about the current frame. Here's an easy program that will return the value of* ***@panel1****.*

@setselection("< frame name > "),
@performkeys("{Dnlevel}"),
@return(@panel1)

.....

I have two Framework questions. First, I cannot get the **Disk:Get File by Name** menu option to work. Every time I try to use it, I get the error message "Unable to open file." I have an IBM PC-AT running under DOS version 3.0. Framework is completely loaded on hard disk, drive C: and I have CONFIG.SYS in the root directory that contains the statement FILES = 30. My frame loads ok from the drive cabinet when I highlight it and press **Return**. Second, how do I get unconditional page breaks when printing spreadsheets? Thanks for any help.

Do you have Framework or Framework II? If you have Framework you must specify the extension of the file when you use ***Disk:Get File by Name****. In Framework II if the extension is not (.FW2) you still must include the extension. If the file has no extension make sure you put a period (.) at the end of the file name.*

As to your second question, there is no way to get unconditional page breaks inside a spreadsheet. You can handle this by either creating an outline with multiple spreadsheets, or ***COPYing (F8)*** *your spreadsheet to a word frame and then entering your page breaks with* ***Edit:Begin New Page****. If you simply want to begin printing a spreadsheet on a new page, you can set this as an option by using the* ***Print:Control Options:Begin Frame on New Page****.*

.....

dBASE III PLUS

Changing the definition of ON ESCAPE from within the called ESCAPE DO doesn't always work. For example, I issue an ON ESCAPE DO < filename > at the start of a .PRG to execute a procedure to CLOSE DATA, RELEASE selected memory variables, and then redefine **Esc** to ON ESCAPE RETURN. An INKEY() loop then waits for the user to press a command key to perform an action. During this process, a record from the active database file is on the screen. Pressing Esc does CLOSE the DATABASE files and RELEASE the memory variables. It is, as well, supposed to display the message "Press **Esc** Again..." for the new definition of ON ESCAPE to return control back to the calling module. What happens sometimes, is that a record from the database file that has been CLOSEd displays, the ON ERROR DO is apparently disregarded and the error messages "Variable not found" or "Alias not found" display with the "Cancel, Ignore, Suspend?" prompt. Pressing any key returns control to the dot prompt and trying to restart the program with DO < file name > returns the error message "Syntax Error." In order to continue, we have to QUIT then reload dBASE III PLUS.

I am working in the following environment IBM PC-XT, with 640K, running under DOS 3.2, and using dBASE III PLUS or dBASE Administrator Version 1.1, standalone or under PC-Net Version 1.1

Any ideas?

I tried what you said and I didn't have any problems. I've never seen a case where a database file that was not in USE got displayed. I don't know of a case where ON ESCAPE does not work as it should. Have you tried to do DISPLAY STATUS at the point where the data from the CLOSEd DATABASE is displayed? I would recommend using SET DEBUG ON and SET ECHO ON and trying to find out what exact instructions are causing the error messages: "Variable not found", "Alias Not Found" and "Syntax Error."

.....

INTERNATIONAL

Using Framework II to Emulate Minitel Terminals

BY ALISTAIR CONNOR AND DIDIER ROCHEREAU,
LA COMMANDE ELECTRONIQUE, FRANCE

In France, one of the most popular ways to access electronic bulletin boards is with Minitel terminals. Minitels are dumb terminals that use Videotex standards for remote terminal control, such as defining cursor positioning, scrolling, and clearing the screen. By using Minitel, you can access vast amounts of information from the thousands of electronic bulletin boards available, get on-line telephone directory listings, and use electronic mail. With the aid of a special modem and terminal emulation driver, you can use Framework II as a Minitel terminal, allowing you to access these electronic services through Framework II telecommunications.

The advantage of using Framework II as a Minitel terminal is that you can continue working in Framework II while sending or receiving information through Minitel. In addition, Framework II can save the information to a text file as it displays information to the screen. This increases your efficiency and may lower your connect-time costs because you won't have to spend time reading or writing down the information while connected. Once the data is saved to a text file, you can filter out the Videotex control characters by using EXTRACT, a utility program that comes with the LCE 22/23 modem card.

Requirements

To use Framework II as a Minitel terminal, you need the LCE 22/23 Carte Multimodem Intelligente and the customized modem device driver (PNB.SYS) developed for Minitel emulation. The LCE 22/23 modem card adheres to the CCITT (Consultative Committee on International Telegraphy and Telephone) V.23 standards, which can send information at 75 baud rate and receive information at 1200 baud rate. La Commande Electronique (LCE), the Ashton-Tate authorized distributor in France, sells the LCE 22/23 modem card with the modem device driver (PNB.SYS) needed for Minitel emulation.

Setup

Setting up Framework II for Minitel emulation requires three separate installations: one for the modem card and device drivers, one for the terminal emulation driver that is located on the Framework II Setup disk (Mintel.TE), and another for the Minitel parameter file (Minitel.TCM) that you create.

Installing the LCE 22/23 Modem Card and Device Driver

Because the LCE 22/23 modem card comes configured for Minitel emulation, installing it simply means placing the card into an empty slot in your computer. It comes configured for COM2:, although you can install it as COM1: by changing the jumpers. For more detailed information regarding the LCE 22/23 modem card, refer to the LCE 22/23 reference manual *Manuel utilisation*. Once the modem card is installed, follow the steps below to install the modem device driver:

1. Copy the file PNB.SYS to the root directory of your hard disk or boot disk. PNB.SYS is located on the disk that comes with the LCE 22/23 modem card.
2. Add the following command line to your CONFIG.SYS file:

   DEVICE = PNB.SYS

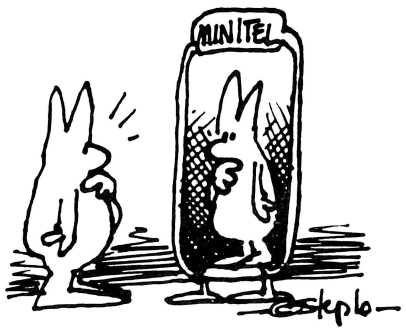
Installing Minitel.TE as a Terminal Emulation Driver

The terminal emulation driver, Minitel.TE, is located on the Framework II Setup disk. If you are not familiar with the Framework II Setup program, follow these steps to install Minitel.TE as an option on the **Apps:Télécommunications: Paramétrer:Terminal** pull-down menu.

1. From the first screen of Setup, select 2, Autres Utilisations Du Programme De Configuration.
2. Select the appropriate option for the location of the Setup file (FWSETUP).
3. From the Menu Principal, select 4, Télécommunications.
4. From the Modification Des Télécommunications menu, select 5, Emulation De Terminaux.
5. A list of five standard terminal emulation drivers is displayed. Choose one that you will not be using, so that the Minitel emulation driver can take its place.
6. Select 6, Le nom de fichier de votre propre interface.
7. Enter Minitel.TE for the driver filename.
8. Follow the screen prompts to save this change.

Creating and Installing the Minitel Parameter File

To create a Minitel parameter file, Minitel.TCM, refer to Figure 1 for the correct settings to enter for **Apps:Télécommunications: Paramétrer:Options de Configuration**. In addition, enter the electronic service telephone number (No à Composer), including an outside-line initialization code, if necessary.



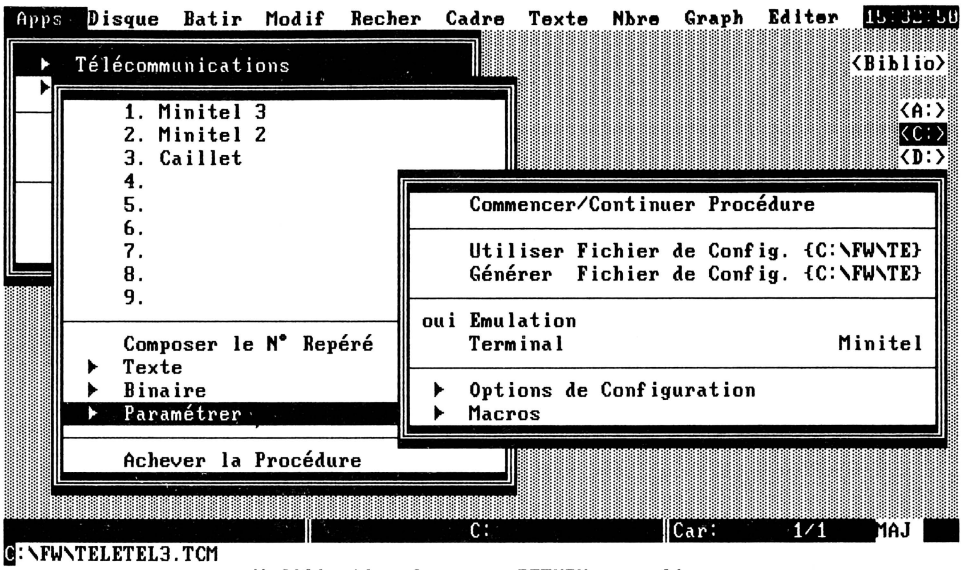


Figure 1.

Once you have verified that the parameter file has the correct settings, save the file and follow the step below to add it to the **Apps:Télécommunications** pull-down menu (Figure 2).

1. From the first screen of Setup, select 2, Autres Utilisations Du Programme De Configuration.
2. Select the appropriate option for the location of the Setup file (FWSETUP).
3. From the Menu Principal, select 4, Télécommunications.
4. From the Modification Des Télécommunications menu, select either 1, 2, or 3 (Elements De Menus) depending on which menu option number you want Minitel to be.
5. For each menu option, there are two entries: one for the menu name and the other for the filename (.TCM). Again, select the appropriate options and enter the menu name and filename.
6. Follow the screen prompts to save this change.

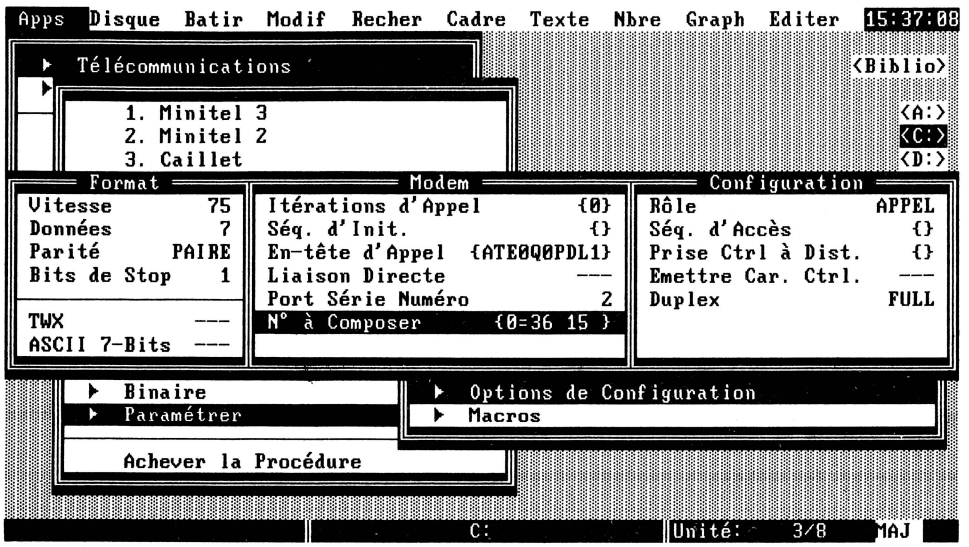


Figure 2.

Usage

Before starting a Framework II session in which you want to emulate Minitel, type

CTTY MINI

from the DOS prompt to initialize the modem device driver PNB.SYS; do not execute this command from **DOSS ACCESS**. Next, load Framework II and select the option for Minitel from the **Apps:Télécommunications** pull-down menu. This loads the Minitel parameter file (Minitel.TCM) and dials the telephone number.

When the Minitel emulation begins, the screen goes blank for about 30 seconds. Once a connection is made, an appropriate main menu screen displays. An example of an entry screen is shown in Figure 3.

Figure 3.

At any point in the telecommunications process, you can return to the Framework II desktop by pressing **Ins** or **Ctrl-Break**. To return to the telecomm session, select **Apps:Télécommunications:Paramétrer:Commencer/Continuer Procédure**.

Saving Data to a Text File

If you want to capture information in a text file, enter a filename for the **Apps:Télécommunications:Texte:Fichier de Mémorisation** option and toggle the **Apps:Télécommunications: Texte:Memoriser dans Fichier** option to "Oui." From that point on, as data is displayed to the screen, it is also written to the specified text file. When you want to stop saving information to the text file, deselect **Apps:Télécommunications:Texte: Memoriser dans Fichier**.

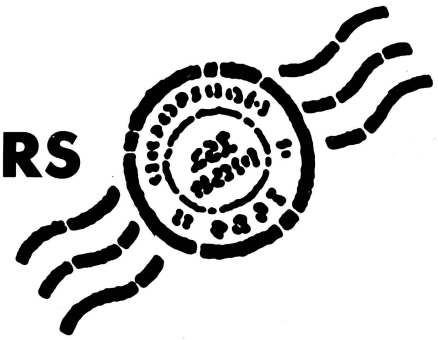
Using EXTRACT

EXTRACT is a utility program that comes with the LCE 22/23 modem card and is used to filter out Videotex control characters from text files. You can execute it from DOS or from DOS Access. To do this, enter

EXTRACT < text filename > **Return**

Using Framework II to emulate Minitel terminals can increase your efficiency by helping you clear off your desk (you won't need the Minitel terminal any more) and by saving you money (you can save information to a text file and read it off-line).

LETTERS



Test for Blank Dates

Dear *TechNotes*:

I have found that if I try to compare date variables against a date which is blank, the test always comes out false (.F.). The following should prove out my findings:

```
. test1 = CTOD("  /  /  ")
. test2 = DATE()
. ? test1 = test2
.F.
. ? test1 = DATE()
.F.
. ? test1 <> DATE()
.F.
. ? test1 <> test2
.F.
. ? DATE()
11/18/86
. ? test1
/ /
. ? test2
11/18/86
. ? test1 # DATE()
.F.
. ? test1 # test2
.F.
```

This problem only seems to appear when testing against a date that prints out as " / / ".

JA

Dear JA:

To reliably test for a date field (or variable) having a null value (no date, just blanks with embedded slashes), use the following expression:

```
DTOC(<date field>) = "  /  /  "
```

If you need to compare for inequality, convert both sides of the conditional expression to character as follows:

```
test1 = CTOD("  /  /  ")
test2 = CTOD("  /  /  ")
? DTOC(test1) = DTOC(test2)
```

Interfacing dBASE III PLUS with External Programs

Dear *TechNotes*:

I am a practicing physician preparing to assemble a software package for use in my office practice on an IBM PC-XT with 640K of RAM and 20 MB of hard disk storage. Please answer the following questions about dBASE III PLUS:

1. When you say dBASE III PLUS requires 256K of RAM, does this figure include DOS 3.2?
2. I will be using dBASE III PLUS (256K) with a full-feature word processor (256K) and a RAM-resident keyboard enhancer (64K). So far, all of this should fit. However, I would also like to run interpreted BASIC applications (or compiled BASIC routines, later on) from within dBASE III PLUS or from within the word processor, if possible. Will everything still fit into 640K?
3. Can I run an interpreted BASIC application from within a memo field, and return to my place within the memo field?
4. How well do each of these word processors work with your memo fields: WordStar, Word Perfect, and Microsoft Word?
5. When I use Config.DB to specify an external word processor for use within memo fields, is the word processor held in RAM (as I suspect it is), or is it read from disk each time a memo field is edited?
6. Can I pass control to other types of programs within a memo field besides word processors?
7. Can I call an assembled application program from dBASE III PLUS?

Thank you very much for your attention.

ASH

Dear ASH:

dBASE III PLUS works on a 256K system running under DOS 2.0 or 2.1; systems running under DOS 3.x require a minimum of 384K to operate.

The configuration of software and memory that you describe is not enough memory space for dBASE III PLUS to operate. Remember that in a 640K system, part of memory is used by DOS, up to 64K in the later versions, so the combination of word processor, dBASE III PLUS and a keyboard enhancer would put you beyond the 640K limit of memory.

When working in a memo field, it is not possible to run any external programs. The only external program that can possibly be associated with memo fields is an external text editor or word processor for use in editing the field's contents.

WordStar is the most widely used editor for memo fields other than the dBASE Word Processor. Word Perfect will possibly work for you, but has not been tested here. Microsoft Word does not work with ASCII text files, and so is unsuitable for use as a memo field editor.

Using an external word processor for your memo field editor (the WP setting in the Config.DB file), each time you edit a memo field the word processor is reloaded into memory from disk. The editor is not retained in memory all of the time.

You can use the LOAD and CALL commands to access assembly-language programs from within dBASE III PLUS. These commands are documented in detail in Using dBASE III PLUS, Chapter 5.

Print Selected Records from a Database Frame

Dear *TechNotes*:

I recently purchased Framework II, which I am enjoying using.

However, I am unable to find the answer to a problem and would appreciate it if you could help me.

My problem concerns printing out a selected name and address from a database and have that single name and address printed on an envelope. Each time I try, the printer begins printing the names and addresses from the the first record name in the database.

I purchased Framework from a local computer store as part of a package when I acquired a new computer which is IBM compatible. No one there seems to know any more about Framework than I do.

I would like to thank you in advance for this favor.

LGW

Dear LGW:

To select specific records in a database frame for printing, you must filter the database frame to select the set of records you want to be active to display. To create a filter, highlight the border of the database frame and press **EDIT FORMULA (F2)**. Then enter a logical expression to filter the database frame. For example, assume you have a field named of "Lname" (for Last Name) and you want to see only those records that contain the word "Smith" in that field. The filter expression you enter into the formula area of the database frame would be:

```
Lname = "Smith"
```

To execute the filter press **RECALC (F5)**. You will then see only those records with a last name of Smith displayed.

Note that filtering a database frame does not delete the records not meeting the filter condition, but rather makes them "invisible" to you (and to Framework II) while the filter is in effect. At this point you can print the database frame, and only the records displayed will appear on the printout.

To redisplay all the records in the database frame, highlight the border of the database frame and then use the **Frames:Open All** menu option.

Cumulative Index

January 1987 through March 1987

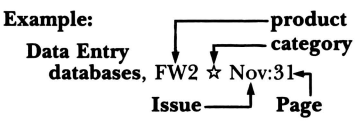
Products:

D2	-	**dBASE II**
D3	-	**dBASE III**
DR	-	**dBASE III Developer's Release**
DP	-	**dBASE III PLUS**
DPA	-	**dBASE III PLUS ADMINISTRATOR**
DB	-	**dBASE Products**
FW	-	**Framework**
FW2	-	**Framework II**
MM	-	**Multimate**
UT	-	**Utilities**

Categories:

☆	-	**Articles**
¥	-	**International**
□	-	**Letters**
○	-	**On-Line Services**
◀	-	**Sample Programs**
§	-	**Usage Tips**

Example:



A

@abs, FW2 ☆ Mar:26
ACCESS, DPA § Feb:50
Add_lf.FW2, FW2 ◀ Jan:46,
See also: Telecommunications
ALTERNATE = in CONFIG.DB, DP ☆ Jan:22
APPEND, DP § Jan:44, DP ☆ Mar:13, FW2 ○ Mar:61
APPEND FROM
SDF, DP § Feb:49, DP ○ Feb:57
Arrays
save and restore, UT § Mar:54
AT(), DP □ Feb:63
AUTOEXEC.BAT, DP ○ Jan:58
@avg, FW2 ☆ Feb:10

B

@beep, FW2 ○ Jan:59
BELL = in CONFIG.DB, DP ☆ Jan:22
BROWSE, DP § Jan:45
EGA 43-line mode, DP ○ Feb:56
WIDTH, DP § Feb:47
BUCKET = in CONFIG.DB, DP ☆ Jan:21

C

CALL, DP ☆ Jan:28, DP ☆ Feb:21, DP ☆ Mar:19
CARRY = in CONFIG.DB, DP ☆ Jan:22
Case folding, DP ¥ Jan:61
CATALOG = in CONFIG.DB, DP ☆ Jan:22
Catalogs, DP □ Jan:64
CENTURY = in CONFIG.DB, DP ☆ Jan:22
@chr(), FW2 ☆ Mar:25
Collation sequence, DP ¥ Jan:61
COLOR = in CONFIG.DB, DP ☆ Jan:22
COMMAND = in CONFIG.DB, DP ☆ Jan:21
Compaq 386, DP § Feb:47
CONFI256.SYS, DP □ Feb:64
CONFIG.DB, DP § Feb:47, DP □ Feb:64, DP □ Mar:70
classes of commands, DP ☆ Jan:20
configuration commands, DP ☆ Jan:21
creating and editing, DP ☆ Jan:20
function keys, DP ☆ Jan:25
set commands, DP ☆ Jan:22
CONFIG.SYS, DP ☆ Jan:40, DP □ Feb:64
BUFFERS =, DP ☆ Jan:19
FILES =, DP ☆ Jan:19
CONFIRM = in CONFIG.DB, DP ☆ Jan:22
CONSOLE = in CONFIG.DB, DP ☆ Jan:22
Converting
numbers to words, DP ☆ Mar:17
COPY, FW2 ☆ Jan:9, DP § Jan:45, FW2 ○ Mar:62
absolute cell references, FW2 ☆ Mar:29
boldface text, FW2 ☆ Mar:29
frames, FW2 ☆ Mar:29
soft hyphens and page breaks, FW2 ☆ Mar:30
STRUCTURE EXTENDED, DP § Mar:52
Cpsep.PRG, DP § Jan:45,
See also: Exporting
CREATE, DP § Mar:52
CREATE LABEL, DP § Feb:48
CREATE SCREEN, DP ☆ Mar:10
Ctrl-Return, FW2 ☆ Jan:9

D

Data entry, FW2 ☆ Feb:10
field ditto, DP ☆ Mar:13
DATE(), DP ¥ Jan:62
Dates
conversions, DP ○ Mar:61
test for blanks, DP □ Mar:69
@dbasefilter, FW2 ○ Feb:58
DBF(), DP □ Jan:64
DCONVERT, DP ○ Jan:57
DEBUG = in CONFIG.DB, DP ☆ Jan:23
DECIMALS = in CONFIG.DB, DP ☆ Jan:23
DEFAULT = in CONFIG.DB, DP ☆ Jan:23, DP ☆ Mar:9
DELETED = in CONFIG.DB, DP ☆J an:23
DELIMITERS = in CONFIG.DB, D P☆ Jan:23
DEVICE = in CONFIG.DB, DP ☆ Jan:23
DIR, DP ☆ Mar:9
@display, FW2 ○ Jan:59
DISPLAY STATUS, DP ○ Mar:63
Ditto.PRG, DP ◀ Mar:14,
See also: Data entry
DOS access, FW2 ☆ Jan:15
DRAG, FW2 ☆ Mar:31
Drive settings, FW2 ☆ Jan:8

E

ECHO = in CONFIG.DB, DP ☆ Jan:23
Editing
display hidden characters, FW2 ☆ Mar:31
documents, FW2 § Jan:46
Undo, FW2 ☆ Mar:31
EDIX, DP ☆ Jan:36
Empty.PRG, DP ◀ Feb:28,
See also: Word wrapping
Error messages in dBASE III PLUS
"*Execution error on +"**, DP ☆ Feb:26

".DBT file cannot be opened", DP § Feb:48
"Alias not found", DP ○ Feb:57, DP ○ Mar:63
"Do you wish to delete field", DP ☆ Mar:10
"Illegal value", DP § Feb:47
"NET 803: Network path not found", DPA ☆ Mar:10
"not readable", DP ☆ Mar:9
"Numeric overflow (data was lost)", DP § Feb:49
"Printer not ready.", DP ☆ Mar:11
"Syntax error in field expression", DP ○ Jan:57
"Syntax error", DP ○ Mar:63
"Variable not found", DP § Jan:45, DP ○ Mar:63
Error messages in Framework II
"Frame badly formatted - unable to read it", FW2 ☆ Mar:25
"Not able to load extended memory driver", FW2 ☆ Jan:10
"Not enough normal memory to use extended memory", FW2 ☆ Jan:11
ESCAPE = in CONFIG.DB, DP ☆ Jan:23
European versions, DP ¥ Jan:60
EXACT = in CONFIG.DB, DP ☆ Jan:23
Exporting
ASCII file, DP § Jan:45
SYLK files, FW2 ☆ Mar:38
EXTEND SELECT, FW2 § Jan:46, FW2 ○ Jan:58
Extended memory, FW2 ☆ Jan:10
External programs
interfacing dBASE III PLUS, DP □ Mar:70

F

File handles, FW2 ☆ Mar:30
Fill.PRG, DP ◀ Mar:54,
See also: CREATE
Filtering, FW2 □ Mar:71
Forms view, FW2 ☆ Mar:32
Framelock, FW2 □ Jan:63
Frames
size, FW2 ☆ Jan:12
FUNCTION = in CONFIG.DB, DP ☆ Jan:25

G

GETS = in CONFIG.DB, DP ☆ Jan:21
Graphics library
save and restore arrays, UT § Mar:54
scaling, UT § Mar:55
Graphing, FW2 ☆ Mar:32
GraphLink
graphics integration, MM ☆ Jan:41

H

HEADINGS = in CONFIG.DB, DP ☆ Jan:23
HELP, FW2 ☆ Jan:3, FW2 ☆ Mar:32
HELP = in CONFIG.DB, DP ☆ Jan:23
@hide, FW2 ☆ Mar:31
HISTORY = in CONFIG.DB, DP ☆ Jan:23

I

ID, FW2 ☆ Jan:4
Importing
dBASE II/III files, FW2 ☆ Mar:30
DELIMITED, DP □ Jan:64
file capture in emulation, FW2 § Jan:47
from a mainframe, DP □ Jan:64
Lotus 1-2-3 files into Javelin, UT ¥ Feb:59
SDF, DP □ Jan:64
SYLK files, FW2 ☆ Mar:38
WordStar, FW2 ☆ Mar:31
INDEX, DP § Jan:44, DP § Feb:48, DP ☆ Mar:9, DPA ☆ Mar:10, DP ☆ Mar:11
Indexing, DP ¥ Jan:62
file size, DP § Jan:44
INKEY(), DP § Feb:48
@inputline, FW2 ○ Mar:61
Installing, FW2 ☆ Jan:4, FW2 ☆ Jan:9
custom drivers, FW2 ☆ Jan:16
@int(), FW2 ☆ Mar:26
INTENSITY = in CONFIG.DB, DP ☆ Jan:23

J

Javelin, UT ¥ Feb:59

K

Keyboard drivers, FW2 ☆ Jan:8
for Tandy 1200 HD, FW2 ☆ Mar:32

L

LABEL FORM, DP ○ Jan:56
Labels, FW2 ☆ Mar:27
LASTDRIVE = in CONFIG.SYS, DP ☆ Jan:40
Library
macros, FW2 ☆ Mar:32
Library.FW2, FW2 ☆ Mar:33
LIST
ALL, DP □ Feb:63
TO PRINT, DP ☆ Mar:11
LOAD, DP ☆ Jan:28, DP ☆ Feb:21, DP ☆ Mar:18
Locate, FW2 ☆ Mar:33
Logical fields and variables, DP ¥ Jan:62
Lotus 1-2-3, FW2 ☆ Feb:10,
See also: Range input

M

Mac.INC, DP ◀ Jan:28,
See also: Screens
Macros, DP ☆ Jan:36, DP ○ Jan:58
auto indentation, FW2 § Mar:55
Mailmerge, FW2 ☆ Mar:27
Make.PRG, DP ◀ Mar:53,
See also: CREATE
MARGIN = in CONFIG.DB, DP ☆ Jan:23
MAXMEM = in CONFIG.DB, DP § Feb:47
Memo fields, DP § Feb:48
Memory, FW2 ☆ Jan:9
overlays, FW2 ☆ Jan:10
MEMOWIDTH = in CONFIG.DB, DP ☆ Jan:23, DP § Feb:47
@menu, FW2 ○ Jan:58
MENU = in CONFIG.DB, DP ☆ Jan:23
@mid(), FW2 ☆ Mar:26
MODE, FW2 ☆ Jan:8, DP § Feb:47
MODIFY COMMAND, DP ☆ Jan:36, DP ○ Jan:57, DP ¥ Jan:62,
See also: Program editors
MODIFY REPORT, DP ○ Jan:57
MODIFY SCREEN, DP ☆ Mar:10
MODIFY STRUCTURE, DPA ☆ Mar:10
MontoNum.FW2, FW2 ○ Mar:61
MOVE, FW2 ☆ Jan:9
MVARSIZ = in CONFIG.DB, DP ☆ Jan:21

N

Ndxsize.PRG, DP ◀ Jan:44
@nextkey, FW2 ○ Jan:59
Norton Editor, DP ☆ Jan:37
Novell, DP § Jan:46
Num2Word.ASM, DP ◀ Mar:19
Numeric coprocessor, FW2 ☆ Jan:8

O

ON ESCAPE, DP ○ Mar:63
Outline view, FW2 ☆ Mar:34
Overlays, FW2 ☆ Jan:10

P

@panel, FW2 ○ Mar:62
Parameters
passing, DP ☆ Feb:22, DP ☆ Mar:18
PATH = in CONFIG.DB, DP ☆ Jan:24
@performkeys, FW2 ○ Jan:58, FW2 ○ Mar:61
Port assignment, FW2 ☆ Jan:4
Print logic, MM ☆ Feb:3
PRINT = in CONFIG.DB, DP ☆ Jan:24
Printer
control, FW2 ☆ Jan:11
driver, FW2 ☆ Jan:4
Printing with dBASE III PLUS
using European versions, DP ¥ Jan:61
@Z, DP ○ Jan:56
Printing with Framework II
a database in dBASE view, FW2 □ Jan:63
driver for NEC 7710, FW2 ☆ Mar:35
Footers, FW2 § Feb:50
format/control, FW2 ☆ Mar:35
frame labels, FW2 ☆ Mar:35
from virtual memory, FW2 ☆ Mar:35
Headers, FW2 § Feb:50
selected records from a database, FW2 □ Mar:71
using attributes, FW2 § Mar:56
Printing with MultiMate

creating endnotes, MM ☆ Mar:48
merging, MM § Mar:57
Printer Action Tables, MM ☆ Feb:3, MM ○ Feb:58
using the WPS format, MM ☆ Mar:48
Program design
modularity, DP ☆ Feb:31
Program editors, DP ☆ Jan:35
WordStar, DP ○ Jan:57
Programming with
dBASE III PLUS
using European versions, DP ¥ Jan:62
PROMPT = in CONFIG.DB, DP ☆ Jan:21
PROTECT, DPA ☆ Mar:10
privileges, DP § Jan:45

R

RAM-resident programs
Sidekick!, DP ☆ Jan:38
Random Access Memory (RAM)
disk, FW2 ☆ Jan:10
Range input, FW2 ☆ Feb:10
Range.FW2, FW2 ◀ Feb:14,
See also: Range input
RangeInput.FW2, FW2 ◀ Feb:15,
See also: Range input
Referecing
regions, FW2 ☆ Mar:36
with English names, FW2 ☆ Mar:36
Referencing
library, FW2 ☆ Mar:36
REINDEX, DP § Jan:44
REPLACE, DP § Feb:49, DPA ☆ Mar:10
ALL, FW2 ○ Mar:61
REPORT FORM, DP ○ Jan:56
menu creation, DP ☆ Feb:30
Restscr.ASM, DP ◀ Jan:32,
See also: Screens
Rptfrm.PRG, DP ◀ Feb:38,
See also: REPORT FORM
RUN, DP ☆ Jan:36, DP ¥ Jan:57
@run(), FW2 ☆ Mar:26

S

SAFETY = in CONFIG.DB, DP ☆ Jan:24
Savescr.ASM, DP ◀ Jan:30,
See also: Screens
Saving
files, FW2 ☆ Mar:37
Sci.PRG, DP ◀ Mar:51,
See also: Scientific notation
Scientific notation, DP § Feb:49, DP § Mar:51
SCOREBOARD = in CONFIG.DB, DP ☆ Jan:24
Screens
AT&T PC 6300, FW2 ☆ Jan:18
color, FW2 ☆ Jan:9
driver, FW2 ☆ Jan:4
IBM Color Graphics, FW2 ☆ Jan:18
saving and restoring to files, DP ☆ Jan:26
Serial number protection, FW2 ☆ Jan:15
SET
ALTERNATE TO, DP § Jan:45
CARRY ON, DP ☆ Mar:13
CATALOG, DP □ Jan:64, DP § Mar:52
DATE, DP ☆ Jan:24
DELIMITERS, DP ☆ Jan:21
DEVICE TO PRINT, DP ☆ Mar:12
DO HISTORY, DP ☆ Jan:24
EXCLUSIVE OFF, DP ☆ Mar:9, DPA ☆ Mar:10
FIELDS, DP ☆ Jan:24
FILTER, DP ☆ Jan:24
FIXED, DP ☆ Jan:24
FORMAT, DP ☆ Jan:24
INDEX, DP ☆ Jan:24
MESSAGE, DP ☆ Jan:24
ODOMETER, DP ☆ Jan:24
ORDER, DP ☆ Jan:24
PATH TO, DP ○ Jan:57
PRINT, DP ☆ Jan:24, DP ☆ Mar:12
PRINTER TO, DP ☆ Jan:24, DP ☆ Mar:12
PROCEDURE, DP ☆ Jan:24
RELATION, DP ☆ Jan:24
STATUS OFF, DP ○ Jan:58
TITLE, DP ☆ Jan:24
UNIQUE ON, DP ☆ Mar:11
@setselection, FW2 ☆ Feb:13
Setup
changing configurations, FW2 ☆ Jan:8
for the first time, FW2 ☆ Jan:4
library of configurations, FW2 ☆ Jan:18
primary hardware, FW2 ☆ Jan:8
secondary hardware, FW2 ☆ Jan:8
Sidekick!, DP ☆ Jan:38
SIZE, FW2 □ Jan:63, FW2 ☆ Mar:37
SORT, DP ☆ Mar:10, DP ☆ Mar:12
Spelling check
macros, FW2 ☆ Mar:28
personal dictionary, FW2 ☆ Mar:28
repeated words, FW2 ☆ Mar:28
Spooling, FW2 ☆ Jan:10
STATUS = in CONFIG.DB, DP ☆ Jan:24
STEP = in CONFIG.DB, DP ☆ Jan:24
SUBST, DP ☆ Jan:40
SUBSTR(), DP □ Feb:64
Substring search, DP □ Feb:63
@sum, FW2 ☆ Feb:10
Sylkexp.FW2, FW2 ◀ Mar:45,
See also: Exporting
Sylkimp.FW2, FW2 ◀ Mar:41,
See also: Importing

T

TALK = in CONFIG.DB, DP ☆ Jan:24
@tan(), FW2 ☆ Mar:26
.TCM files, FW2 ☆ Mar:25
TEDIT = in CONFIG.DB, DP ☆ Jan:22, DP ○ Jan:57
Telecommunications
auto line feeds, FW2 § Jan:46
Terminal Emulation, FW2 ¥ Mar:64
TimeFrame, FW2 □ Jan:63
Toshiba 1100 +
Dnlevel/Uplevel, FW2 □ Feb:65
@trace(), FW2 ☆ Mar:26
TRANSFORM(), DP ○ Jan:56
TYPEAHEAD = in CONFIG.DB, DP ☆ Jan:24

U

Uninstall, FW2 ☆ Jan:9
UNIQUE = in CONFIG.DB, DP ☆ Jan:24
Upgrading
from version 1.0, FW2 ☆ Jan:5

V

@var, FW2 ☆ Feb:10
VIEW, FW2 □ Jan:63
VIEW = in CONFIG.DB, DP ☆ Jan:24

W

@while, FW2 ☆ Jan:15
Word wrapping, DP ☆ Feb:21
WordStar, DP ☆ Jan:39
Wordwrap.ASM, DP ◀ Feb:24,
See also: Word wrapping
Wordwrap.BIN, DP ◀ Feb:21,
See also: Word wrapping
WP = in CONFIG.DB, DP ☆ Jan:22
Wraptest.PRG, DP ◀ Feb:27,
See also: Word wrapping
@writetextfile, DP ☆ Mar:26

Z

ZAP, DP § Mar:52
ZOOM, FW2 ☆ Jan:9, FW2 ☆ Jan:15, FW2 ☆ Mar:37

QUARTERLY/*TECHNOTES* CHANGE OF ADDRESS

If you're moving, don't forget to tell us so that we can be sure your subscription to the *Ashton-Tate Quarterly* and/or ***TechNotes*** is uninterrupted. To process a change of address, we *must* have your previous mailing address (including the zip code) and your new address.

My new address is:

Name/Dept

Company Name/Add'l Destination Info

Mailing Address

City State Zip Country

Please affix your current mailing label(s) here. (If the label is not available, be sure to attach your old address.)

Affix your current **Ashton-Tate Quarterly** mailing label here	Affix your current ***TechNotes*** mailing label here

Return to **Ashton-Tate Periodicals, PO Box 3729, Escondido, CA 92025-0929**

To change your address on the Ashton-Tate Product Mailing List, necessary to receive upgrade and other important information about your Ashton-Tate product(s), you **must** send a separate change of address to: Ashton-Tate Customer Service, 20101 Hamilton Ave., Torrance, CA 90502-1319.

CUSTOMER SERVICE FOR TECHNOTES

Customer service and order fulfillment for ***TechNotes*** are handled by Publishers Aide, a professional magazine service located in Escondido, California. All subscription inquiries and orders should be sent directly to: **Ashton-Tate Periodicals, PO Box 3729, Escondido, CA 92025-0929**. The following general information on policy and procedures is provided for your information when you have any questions about your subscription or services.

Subscription Inquiries: Since inquiries frequently need to be researched, it is best to write if you have questions about your subscription. To resolve subscription inquiries as quickly as possible, you should include all pertinent information; for example, if you haven't received any subscription issues, you should send the fulfillment service a copy of your cancelled check or credit card charge, with the requested subscription address. If you prefer, you may call one of the following phone number from 9 a.m. to 5 p.m. Pacific Standard Time: In the U.S., call 800-828-2514 (California residents call 800-331-4164). Outside the U.S., call 619-485-6535. When communicating with the fulfillment service, it is best to have a mailing label from your subscription available. The top line of the label is your unique subscriber code. When you give the code to the customer service representative, he or she will be able to locate your record. If you do not have a label available, the representative will be able to search for your record when you provide your zip code, the addressee, and your street address. On the far right of the first line of your mailing label, you can find when your subscription will expire. The abbreviated date identifies the month and year of the last issue of your subscription.

Change of Address: If you move, remember to send us your change of address; bulk mail is not automatically forwarded by the Post Office. To process a change of address, we must have both your new address and your previous address. Because subscription records for ***TechNotes*** and the *Ashton-Tate Quarterly* are maintained on separate databases, you must tell the fulfillment service to change the records for your subscriptions if you subscribe to both magazines and want to change your address on both files. See the Change of Address form in this issue.

New Orders: Please use the order from in the issue to order, and allow 6 to 8 weeks to receive your first issue. Each issue of ***TechNotes*** is mailed twice to ensure that new subscribers receive their first issue promptly. We cannot honor requests to back-date subscription requests.

Missed or Damaged Issues: If you haven't received your magazine by the 20th of the month or if you receive a damaged copy, let us know so that we can replace it for you and/or correct your mailing address.

Renewal Orders: You will be sent a notice about renewing your subscription automatically, but you can write or call to renew without the notice. If you renew your subscription without using the prepared renewal notice, be sure to tell us you are renewing and the address of the current subscription (or better yet, a copy of your mailing label from the current subscription). That way, you'll eliminate the possibility of our starting a new subscription rather than extending a current one.

Questions or Comments: If you have any unresolved subscription problems, you should write directly to: Brenda Johnson-Grau, ***TechNotes***, Ashton-Tate, 20101 Hamilton Avenue, Torrance, CA 90502-1319.

Other Ashton-Tate Matters: Publishers Aide *only* handles *Ashton-Tate Quarterly* and ***TechNotes*** subscription orders. Their operators are not trained to handle questions about other Ashton-Tate products nor can they transfer your call to Ashton-Tate. If you have any product-related questions or wish to order products or services from Ashton-Tate, avoid delay by sending them directly to **Ashton-Tate, 20101 Hamilton Avenue, Torrance, CA 90502-1319.**

TechNotes Reader Survey

We would like to make sure that ***TechNotes*** continues to serve the needs of our readership. Please take the time to fill out this survey and send it to this address:

TechNotes
ATTN: Brenda Johnson-Grau
20101 Hamilton Avenue
Torrance, CA 90502-1319

1. Which Ashton-Tate product(s) do you use? (Check all that apply.)

   ___ dBASE II ___ dBASE III ___ Developer's Release

   ___ dBASE III PLUS ___ Framework ___ Framework II

   ___ dBASE III PLUS LAN Pack ___ MultiMate

2. How long have you used Ashton-Tate products? ___ yrs ___ mos

3. How long have you subscribed to ***TechNotes*** ? ___ yrs ___ mos

4a. Do you use a computer in your work? ___ Yes ___ No

b. If so, what is your job title?

   ___ Developer ___ Educator

   ___ Consultant ___ Data Entry Clerk

   ___ Programmer/Analyst ___ Other (please specify)_____

   ___ Engineer

5. How much computer experience do you have? ___ yrs ___ mos

6. Do you keep back issues of ***TechNotes*** for reference? ___ Yes ___ No

7. How many people, besides you, read your copy of ***TechNotes***? _____

8a. Do you find ***TechNotes*** to be a useful source of technical support?

___ Yes ___ No If No, please explain _____

_____

_____

b. Please rank the following from 1 to 6 in order of usefulness, 1 being most useful and 6 least useful.

   ___ Articles ___ Anomalies ___ On-Line

   ___ Sample Programs ___ Letters

   ___ Usage Tips ___ Update Policies

c. Which articles have been most helpful?

_____

_____

_____

9. Would you be interested in having ***TechNotes*** programs available on SofStrips? ___ Yes ___ No

10. Would you be interested in reading technical information about Ashton-Tate Macintosh products? ___ Yes ___ No

BACK ISSUES ORDER FORM

Order Back Issues! Add to your library of dBASE II, dBASE III PLUS, Framework II, and MultiMate technical reference material. To order your Back Issue(s), fill out this form, enclose your payment, and return to:

***TechNotes* Back Issues**
Ashton-Tate Software Support Center
Attn: Kathy Smith
20101 Hamilton Ave.
Torrance, CA 90502-1319

Name

Company

Address

City State/Province Zip Code

Country Phone Number ( )

All 1985 Back Issues are $10.00 each. All 1986 and 1987 Back Issues are $7.00 each. All prices include postage and tax.

The following issues are currently available. (Those crossed out are out of print.) Check those you wish to order:

( ) FEBRUARY 1985
~~( ) MARCH 1985~~
~~( ) APRIL 1985~~
( ) MAY 1985
( ) JUNE 1985
( ) JULY 1985
~~( ) AUGUST 1985~~
~~( ) SEPTEMBER 1985~~
( ) OCTOBER 1985
( ) NOVEMBER 1985
( ) DECEMBER 1985 (includes 1985 Annual Disk and 1985 Cumulative Index)
( ) JANUARY 1986
( ) FEBRUARY 1986
( ) MARCH 1986
~~( ) APRIL 1986~~
( ) MAY 1986
( ) JUNE 1986
( ) JULY 1986
( ) AUGUST 1986
( ) SEPTEMBER 1986
( ) OCTOBER 1986
( ) NOVEMBER 1986
( ) DECEMBER 1986
( ) JANUARY 1987
( ) FEBRUARY 1987

$_____ TOTAL AMOUNT ENCLOSED

( ) VISA ( ) MasterCard # _____

Expiration Date _____ Signature _____

ALL ORDERS MUST BE PREPAID. MAKE CHECKS PAYABLE TO ASHTON-TATE (IN U.S. DOLLARS DRAWN ON U.S. BANK). PLEASE ALLOW SIX (6) WEEKS FOR DELIVERY OF YOUR ORDER.

NOTE: Information contained in ***TechNotes*** is current as of the issue in which it appears. Therefore, some information found in the Back Issues may no longer be valid (such as Update Policies, etc.).

SUBSCRIBE TODAY

Don't miss a single issue of Ashton-Tate's hottest source of technical know-how! Each monthly issue of TechNotes provides you with the most current, accurate technical information available from the Software Support Center on all dBASE, Framework, and MultiMate products. Call our toll-free numbers to subscribe: **(800) 331-4164** (in California) or **(800) 828-2514** (residents of other states). In Canada, call **(619) 487-3212.**

☐ Bill me:
Amount Enclosed
☐ $50 U.S.
☐ $65 Canada & Mexico
☐ $75 Other countries

☐ Form of payment Enclosed
☐ Check (Make payable to Ashton-Tate in U.S. dollars drawn on a U.S. bank)
☐ Visa ☐ MasterCard

Expiration Date:

Signature:

☐ New Subscription ☐ Renewal

Name

Address

City

State/Province

Zip/Postal Code

Country

Phone Number

Which product do you use? ☐ dBASE II (D2) ☐ dBASE III (D3) ☐ dBASE III PLUS (DP) ☐ Framework (FW) ☐ MultiMate (MM)

Please allow 30 to 60 days for delivery of your first issue. 1000

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BACK ISS

Order Back Issues! Add t
reference material. To ord

***TechNotes* Back Issues**
Ashton-Tate Software Su
Attn: Kathy Smith
20101 Hamilton Ave.
Torrance, CA 90502-1319

Name

Company

Address

City

Country

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 66 TORRANCE, CA

POSTAGE WILL BE PAID BY ADDRESSEE

TechNotes Subscription Service

P.O. Box 3729
Escondido, CA 92025-0929

All 1985 Back Issues are $[illegible] each. All 1986 and 1987 Back Issues are $7.00 each. All prices include postage and tax.

The following issues are currently available. (Those crossed out are out of print.) Check those you wish to order:

( ) FEBRUARY 1985
~~( ) MARCH 1985~~
~~( ) APRIL 1985~~
( ) MAY 1985
( ) JUNE 1985
( ) JULY 1985
~~( ) AUGUST 1985~~
~~( ) SEPTEMBER 1985~~
( ) OCTOBER 1985
( ) NOVEMBER 1985
( ) DECEMBER 1985 (includes 1985 Annual Disk and 1985 Cumulative Index)
( ) JANUARY 1986
( ) FEBRUARY 1986
( ) MARCH 1986
~~( ) APRIL 1986~~
( ) MAY 1986
( ) JUNE 1986
( ) JULY 1986
( ) AUGUST 1986
( ) SEPTEMBER 1986
( ) OCTOBER 1986
( ) NOVEMBER 1986
( ) DECEMBER 1986
( ) JANUARY 1987
( ) FEBRUARY 1987

$_____ TOTAL AMOUNT ENCLOSED

( ) VISA ( ) MasterCard # _____

Expiration Date _____ Signature _____

ALL ORDERS MUST BE PREPAID. MAKE CHECKS PAYABLE TO ASHTON-TATE (IN U.S. DOLLARS DRAWN ON U.S. BANK). PLEASE ALLOW SIX (6) WEEKS FOR DELIVERY OF YOUR ORDER.

NOTE: Information contained in ***TechNotes*** is current as of the issue in which it appears. Therefore, some information found in the Back Issues may no longer be valid (such as Update Policies, etc.).

TechNotes
P.O. Box 3729
Escondido, CA 92025-0929

Address Correction Requested

BULK RATE
U.S. POSTAGE
PAID
PERMIT NO. 27
CULVER CITY, CA